STOCHASTIC LANCHESTER AIR-TO-AIR CAMPAIGN MODEL

METHODS USED TO GENERATE MODEL OUTPUTS AND A USERS GUIDE: 2007

REPORT PA603T2

Robert V. Hemm
David A. Lee
Jeremy M. Eckhause
John A. Dukovich

LMI
GOVERNMENT CONSULTING

MAY 2007

LMI

Stochastic Lanchester Air-To-Air Campaign Model
Methods Used to Generate Model Outputs and a User's Guide: 2007

PA603T2/MAY 2007

Executive Summary

This report documents the latest version of the Stochastic Lanchester Air-to-Air Campaign Model (SLAACM), developed by LMI for the Tactical Air Forces Division of the Program Analysis and Evaluation Directorate of the Office of the Secretary of Defense (OSD PA&E TACAIR).

SLAACM is a fast and flexible model designed for analysts who need to consider many combat scenario options quickly and who wish to have indications of the uncertainties in military outcomes. SLAACM models the defensive counter-air battle, including order of battle optimization by both Red attackers and Blue defenders. The current version can address issues of fighter combat effectiveness, Blue-side battle management capabilities, time-phasing of offensive and defensive forces, use of indigenous defensive forces, and basing options for deployed defensive forces. Unique, efficient mathematical algorithms and fast integer programming tools allow SLAACM to generate optimized results including statistical uncertainties for numerous types and large numbers of combatants for campaigns of many days, with sufficient speed to allow statistical variation of input parameters. Included tables and charts display the day-by-day development of the campaign. In addition, the Microsoft Excel implementation of the model allows wide flexibility for storing and displaying results in user-preferred formats.

This report includes descriptions of the fundamental mathematics and analytical structure of SLAACM and a user's guide. The report also includes discussions of SLAACM extensions to address issues such as weapons expenditure (missile counting) and engagements involving radar tracking and electronic countermeasures.

Planned future work includes application of SLAACM methodology to optimized offensive air campaigns, including suppression of enemy air defense and the impact of electronic warfare and low observable technology.

The authors wish to thank the Director of OSD PA&E, Mr. Bradley Berkson for his support of the development of SLAACM.

Contents

Figures

Tables

Chapter 1
Introduction

The Stochastic Lanchester Air-to-Air Campaign Model (SLAACM), developed by LMI, is a parsimonious probabilistic defensive counter-air model intended to be a flexible tool for analysts who need to consider many cases quickly in the PC environment and who wish to have indications of the uncertainties in military outcomes. SLAACM outputs in benchmarking tests compare closely with those of some large-scale simulation models, but SLAACM is intended to complement, rather than replace, such models. SLAACM typically runs a fully optimized 10-day campaign on a standard PC in less than 2 minutes. This report describes the analytical and statistical methods used to generate SLAACM outputs. It also gives detailed instructions for operating SLAACM, with illustrative examples.

SLAACM Overview

SLAACM treats campaigns between two opponents, called Red and Blue. The Red side uses fighter-escorted bombers to attack assets defended by the Blue side's fighters. In each day's operations, the Red side determines optimal allocations of its fighters and bombers to attack packages, maximizing a payoff function that considers the bombers' payload and weapon-delivery accuracy, the value to Red of destroying Blue fighters, and the penalty to Red of losing fighters and bombers.

Blue's fighters respond. Each Blue fighter can be "smart" or not. Smart fighters can determine the makeup of oncoming Red attack packages before engaging Red and can coordinate their operations to make optimal defenses, maximizing a payoff function that considers the value to Blue of destroying Red fighters and bombers, and the penalty to Blue of losing a fighter.

Blue fighters that are not smart encounter Red's packages randomly. An adjustable parameter varies the effectiveness of Blue's battle management for not-smart fighters, reducing the probability of engagements below that implied simply by the numbers of attacking packages and defending flights.

SLAACM does not use iterative simulation. Rather, it calculates the outcome probabilities of Blue fighters versus Red fighter-plus-bomber packages using analytic probability algorithms. With those results, SLAACM determines statistics of the results of a day's combat. These are used to determine Red and Blue orders of battle for the next day's operations.[1]

[1] The term "day" here refers to a single set of campaign sorties. There could be more than one such set on a single calendar day or there could be calendar day gaps between sets.

SLAACM tracks the total tons of bombs delivered, as well as the tons of smart bombs (guided munitions with higher destructive capability per ton). SLAACM also tracks losses to both sides, which Blue aircraft killed which Red aircraft and vice versa, and standard deviations of Blue and Red losses.

SLAACM's worksheets and analytic structure provide complete audit trails for the input parameters used in the calculations. The kill rate ratios used in the calculations are derived directly from input loss ratios that are displayed in the model. In classified versions of SLAACM, the loss ratios are based on documented historical data, simulations, and military expertise. Further, the analytic nature of the model ensures that the effects of parameter changes are consistently and individually reflected in the output and are not convolved with other parameter effects as happens in iterative simulations. These features can be quite helpful to clarify the implications of the model's results.

REPORT ORGANIZATION

This report is organized as follows:

- Chapters 2 through 5 describe how SLAACM works. Specifically, those chapters, respectively, describe how the model is used to
  - calculate air-to-air engagement results,
  - calculate bomber effectiveness,
  - specify the Red and Blue payoff logic and calculate optimal attacks and defenses, and
  - calculate campaign results.
- Chapter 6 is a users manual for operating SLAACM.

The appendixes contain supplemental information of interest for combat modeling:

- Appendix A discusses the mathematics of the fast engagement analysis algorithm.
- Appendix B addresses the use of NASA-developed Markov tools for auxiliary engagement analysis and scenario development.
- Appendix C discusses SLAACM alternatives.
- Appendix D discusses advanced probabilistic engagement models.
- Appendix E discusses modeling of electronic attack

- Appendix F discusses a mathematical analysis of phased attack and defense

Chapter 2
How SLAACM Works: Engagement Models and Calculations

SLAACM does not simulate; rather, it makes complete calculations of outcome probability distributions for specific probabilistic engagement models. This is a significant difference between SLAACM and other campaign models, so it is appropriate to begin our discussion of how SLAACM works by explaining that approach.

THE BASIC M VS. N PROBABILISTIC ENGAGEMENT MODEL

The heart of SLAACM's calculations is a set of probabilistic models of air-to-air engagements. The simplest engagement model used in SLAACM is the M vs. N probabilistic engagement model. We describe that model in considerable detail, to clarify some fundamental aspects of SLAACM's operation.

In the basic M vs. N probabilistic model, M Blue aircraft engage N Red aircraft. Throughout the engagement, the time between kills by each Blue aircraft is taken to be identically and independently distributed exponentially with parameter k_b, and the time between kills by each Red aircraft is assumed to be identically and independently distributed exponentially with parameter k_r. Specifically,

$$\tau_r \sim k_r e^{-k_r \tau_r} \qquad \text{[Eq. 2-1]}$$

and

$$\tau_b \sim k_b e^{-k_b \tau_b}, \qquad \text{[Eq. 2-2]}$$

where the "~" means "distributed as" and τ_r and τ_b are, respectively, the time between kills made by each Red aircraft and the time between kills made by each Blue aircraft.

If the time between kills by an engaged aircraft is exponentially distributed with parameter k, then the probability that the aircraft makes a kill in a time interval $(t, t + \delta t)$ is $k\delta t + O(\delta t^2)$,[1] independent of t. (This property is sometimes called the "memoryless" property of exponential distributions.)

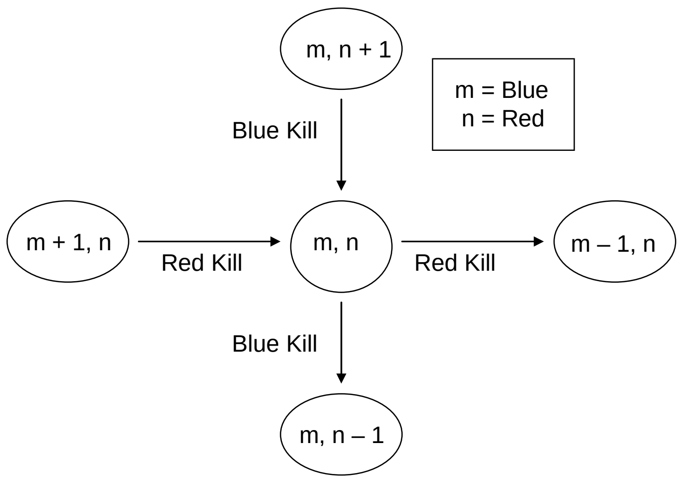
---

[1] $O(\delta t^2)$ signifies "higher order terms" of order δt^2 and higher, which are very small and can be ignored.

If two such aircraft are present, then by virtue of the assumption that their times between kills are statistically independent, the probability that the pair makes a kill in the interval $(t, t + \delta t)$ is $k\delta t + k\delta t + O(\delta t^2) = 2k\delta t + O(\delta t^2)$. If m such aircraft are present, the probability that the set of them makes a kill in the interval $(t, t + \delta t)$ is $mk\delta t + O(\delta t^2)$.

If, at a given time during the engagement, m Blue aircraft and n Red aircraft are present, we will say that the engagement is in state (m, n). Figure 2-1 diagrams the possible transitions that the engagement can make into and out of state (m, n).

Figure 2-1. Engagement State Transition Diagram



Now let $P_{m,n}(t)$ be the probability that the engagement is in state (m, n) at time t. Let us consider the probability $P_{m,n}(t + \delta t)$. The engagement can reach state (m, n) at time $(t + \delta t)$ in just three ways:

- The engagement was in state (m, n) at time t, and no kills happened during the interval $(t, t + \delta t)$.
- The engagement was in state (m, n + 1) at time t, and the set of Blue aircraft made a kill during the interval $(t, t + \delta t)$.
- The engagement was in state (m + 1, n) at time t, and the set of Red aircraft made a kill during the interval $(t, t + \delta t)$.

By the assumptions that all the kill events are independent, these three events are statistically independent. Accordingly, the probability that one of them occurs is the sum of their individual probabilities, so that, neglecting terms of $O(\delta t^2)$,

$$P_{m,n}(t + \delta t) = P_{m,n}(t)\left[1 - mk_b\delta t - nk_r\delta t\right] + P_{m,n+1}(t)mk_b\delta t + P_{m+1,n}(t)nk_r\delta t \text{ .[Eq. 2-3]}$$

Subtracting $P_{m,n}(t)$ from each side of Equation 2-3, dividing by δt, and taking the limit as $\delta t \to 0$ gives

$$\dot{P}_{m,n}(t) = -(mk_b + nk_r)P_{m,n} + mk_b P_{m,n+1} + nk_r P_{m+1,n}\,. \qquad \text{[Eq. 2-4]}$$

With the result (Equation 2-4), we can write down a general initial value problem describing the evolution of the statistics of an M vs. N engagement, in which the Blue forces break away when they have fewer than b_{min} aircraft, and the Red forces break away when they have fewer than r_{min} aircraft:

$$\begin{aligned}
&\dot{P}_{m,n} = -(mk_b + nk_r)P_{m,n} + mk_b P_{m,n+1} + nk_r P_{m+1,n}\,, \\
&\qquad \forall\, b_{min} \le m \le M \text{ and } r_{min} \le n \le N; \\
&\dot{P}_{0,n} = nk_r P_{b\,min,n}\,,\ \ r_{min} \le n \le N \\
&\dot{P}_{m,0} = mk_b P_{m,r\,min}\,,\ \ b_{min} \le m \le M \\
&P_{m,n}(0) \equiv 0 \text{ except } P_{M,N}(0) = 1
\end{aligned} \qquad . \qquad \text{[Eq. 2-5]}$$

(The probabilities that the system has more than M Blues, or more than N Reds, are of course always zero.)

The initial value problem (Equation 2-5) presents a system of linear ordinary differential equations. The number of equations is equal to $(M - b_{min} + 1)(N - r_{min} + 1) + (M - b_{min} + 1) + (N - r_{min} + 1)$. The equations for the probabilities of the absorbing boundary states, $P_{0,n}$ and $P_{m,0}$, decouple from those for the transient states (those whose probabilities tend to zero for long times), however. Therefore, Equation 2-5 may be treated by solving only the $(M - b_{min} + 1)(N - r_{min} + 1)$ equations for the transient states and computing the probabilities of the absorbing boundary states by integration.

In any case, Equation 2-5 offers no difficulties to numerical solution. Moreover, introducing the non-dimensional time parameter $\tau \equiv k_r t$ allows us to write Equation 2-5 in a form that involves only the non-dimensional parameter Kill Rate Ratio = KRR = $\kappa = k_b/k_r$:

$$\begin{aligned}
&\dot{P}_{m,n} = -(m\kappa + n)P_{m,n} + m\kappa P_{m,n+1} + nP_{m+1,n}\,, \\
&\qquad \forall\, b_{min} \le m \le M \text{ and } r_{min} \le n \le N; \\
&\dot{P}_{0,n} = nP_{bmin,n}\,,\ \ r_{min} \le n \le N \\
&\dot{P}_{m,0} = m\kappa P_{m,rmin}\,,\ \ b_{min} \le m \le M \\
&P_{m,n}(0) \equiv 0 \text{ except } P_{M,N}(0) = 1
\end{aligned} \qquad . \qquad \text{[Eq. 2-6]}$$

The state probabilities in Equation 2-6 are functions of τ, not t, and so are different functions from those giving the state probabilities as functions of t. Also the derivatives in Equation 2-6 are with respect to τ, not t. In the interest of unclut-

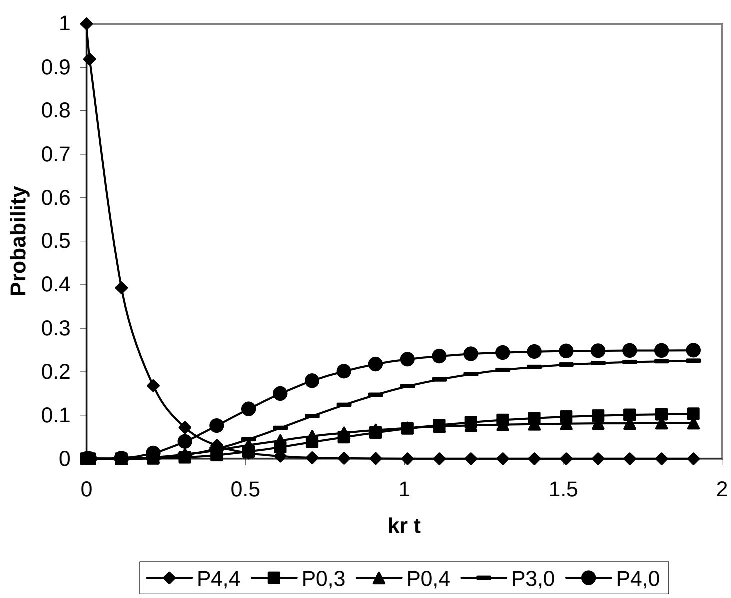
tered notation, we will not introduce new symbols for these new functions; rather, in Equation 2-6 and in all following parts of this chapter, the probabilities are functions of τ, and dots denote differentiation with respect to τ.

Writing the initial value problem for the M vs. N probabilistic engagement in non-dimensional form gives useful insights into the nature of the problem's solutions. First, because the only parameter appearing in Equation 2-6 is κ, we see that the probabilities $P_{m,n}$, which in general depend on time t and the two parameters k_r and k_b, can always be written as functions of the non-dimensional time τ and just one parameter, the kill-rate ratio κ.

Second, the fact that the solutions of the initial value problem depend only on κ and τ leads to the conclusion that the limiting values of the probabilities as time tends to infinity (which of course is also the limit as τ tends to infinity) are functions only of the single parameter κ.

Figure 2-2 illustrates these ideas. It shows the evolution of selected probabilities, obtained by numerical integration for a 4 vs. 4 engagement with $k_b/k_r \equiv \kappa = 1.46$, and in which the combatants fight to annihilation.[2]

Figure 2-2. Evolution of State Probabilities



To find the value of one of the probabilities shown in Figure 2-2 at a given time t for a given engagement, one would find the value of τ corresponding to that t by computing $\tau = k_r t$ and reading the desired value from the figure. For example, at t = 20 time units in an engagement in which the mean time for the Red aircraft to make a kill is 180 time units, the value of τ is 20/180 ≈ 0.11; thus the desired value of $P_{4,4}$ is 0.4.

[2] This engagement also includes absorbing states (2, 0), (1, 0), (0, 2), and (0, 1) that are not shown in the figure.

We will be more interested in the long-time limiting values of the probabilities. Figure 2-2 indicates that for $\tau > 2$, the absorbing boundary states have essentially reached their long-time limiting values. The numerical solution confirms that for $\tau > 2$, the total probability of the absorbing boundary states is larger than 95 percent. The physical time required to reach the long-time limit will of course vary with the value of the dimensional parameter k_r. But the values of the probabilities depend only on the single, non-dimensional parameter κ.

Relating Loss Data to Kill Rate Ratios

The engagement models underlying SLAACM, like all Lanchester-type models, employ kill rates for the opposing forces. In deterministic Lanchester models, the parameters are kills per firer per unit time. In stochastic Lanchester models like SLAACM, the parameters are the reciprocals of mean times between kills by a single firer. For simplicity, we refer to parameters of stochastic Lanchester models as "kill rates."

Observed engagement data, from simulations or from observations of actual combat, are rarely, if ever, rate data. Rather, they are typically loss ratios for specific M vs. N engagements. To apply empirical loss ratio data to rate-based models, we need a method to convert from loss ratio data to kill rate data. To do this, we make the significant assumption that the long-time limits of SLAACM's engagement models are appropriate models for the observations.

Thus, with the basic M vs. N engagement model, we assume that the observed engagements end only with one side's reaching its specific breakaway condition—a certain number of losses, or annihilation—rather than with a clock-time-dependent condition such as fuel exhaustion, or an event-dependent condition such as exhaustion of munitions. (More advanced engagement models, described in Appendixes B and D, do account for missiles, and can be used in SLAACM.)

As noted above, the non-dimensional kill rate ratio parameter, κ, is the single parameter of interest in the basic engagement model. To infer a value of κ from loss ratio data for an M vs. N engagement, we find the value of κ for which the long-time limiting value of the ratio of expected Blue loss to expected Red loss in the basic M vs. N probabilistic model is equal to the observed loss ratio. In this way, we infer a non-dimensional quantity, κ, from a non-dimensional quantity, the observed loss ratio.

Suppose, for example, that the available data are for a 2 vs. 2 engagement to annihilation and that they show a loss ratio of 1 Blue to 10 Reds. Calculations show

that, for the basic probabilistic engagement model, a kill rate ratio κ of 8.28 makes the ratio of expected Blue loss to expected Red loss equal to 1:10.[3]

Although we usually use a finite-step iterative method to compute long-time limits directly, it is of course possible to find long-time limits with time-based methods, such as numerical integration. In time-based Markov calculations, the time required to reach the long-time limit will depend on the value chosen for k_r, where k_b is equal to κ times k_r. For example, analysis shows that the kill rate ratio of 8.28 results in a 1:10 loss ratio in approximately 0.22 time units for $k_r = 1$, in roughly 2.2 time units iterations for $k_r = 0.1$, and in about 22 time units for $k_r = 0.01$. The 0.2, 2, and 22 time units represent adequate model time to bring the solution to the long-time limit, for their corresponding kill rates. Thus, in a time-unit-based model, selection of the individual rates is arbitrary as long as the ratio κ is preserved and the calculations continue to a time on the order of twice the inverse of the sum of the parameters k_b and k_r. (Appendix A describes the general analytic method that produces these numerical results.)

Another major assumption in our engagement modeling is that the kill rate ratio is independent of the Blue:Red aircraft ratio. We assume that kill rate ratios derived from 4 vs. 8 data can be applied directly to 4 vs. 4 and other engagement combinations. A corollary assumption is that kill rate ratios are independent of the state of an engagement. These are significant and powerful simplifying assumptions, which we are always eager to test with new combat data.

Available data from TAC BRAWLER engagement simulations give expected values of Red losses and Blue losses. These data, some of them classified, are available for several Red and Blue aircraft types of interest. Government subject matter experts have scaled these results to model additional aircraft combinations. A SLAACM utility uses bisection iteration and the long-time limit algorithms (discussed below and in Appendix A) to calculate the values of κ from the loss ratios for each of the Red vs. Blue combinations. The calculations are repeated until successive iterations agree to four significant figures.

Calculations for the Long-Time Limit

If the engagement continues to completion—that is, until the probabilities that the system is not in one of the absorbing boundary states (0, n) or (m, 0) are negligibly small—then the probabilities of these states are functions only of κ and the initial values M and N. These conditions allow significant simplification in the calculation of state probabilities. One can obtain values of the long-time limits of the probabilities of absorbing boundary states by a simple, finite-step iteration.

[3] The kill rate ratio reflects the performance of individual Blue vs. Red pairs and is numerically equal to the loss ratio for a 1 vs. 1 engagement. A 10:1 kill rate ratio would be required to produce a 1:10 loss ratio for a 1 vs. 1 engagement, but only an 8.28:1 kill rate ratio is necessary for the same loss ratio in a 2 vs. 2 engagement, because all available aircraft are contributing to each kill.

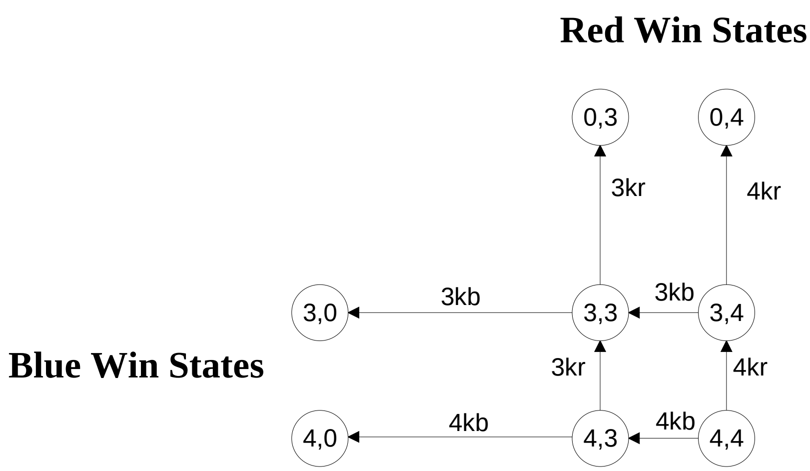
This speeds up the calculations tremendously, compared to time-based Markov methods, and is a key to SLAACM utility.

An Example

As an example of the use of the initial value problem (Equation 2-5), we consider four Blues engaging four Reds. Both sides break away on sustaining 2 losses, that is, $b_{min} = r_{min} = 3$.

Figure 2-3 illustrates the states in the engagement and the transitions between them. In that figure, the absorbing boundary states are labeled as having 0 aircraft for the side that has broken away.

Figure 2-3. Engagement Diagram



For this case, the general initial boundary value problem (Equation 2-6) take this specific form:

$$
\begin{aligned}
\dot{P}_{4,4} &= -(4\kappa + 4)P_{4,4} \\
\dot{P}_{4,3} &= -(4\kappa + 3)P_{4,3} + 4\kappa P_{4,4} \\
\dot{P}_{3,4} &= -(3\kappa + 4)P_{4,3} + 4P_{4,4} \\
\dot{P}_{3,3} &= -(3\kappa + 3)P_{3,3} + 3P_{4,3} + 3\kappa P_{3,4} \\
\dot{P}_{4,0} &= 4\kappa P_{4,3} \qquad , \\
\dot{P}_{3,0} &= 3\kappa P_{3,3} \\
\dot{P}_{0,3} &= 3P_{3,3} \\
\dot{P}_{0,4} &= 4P_{3,4} \\
P_{m,n}(0) &\equiv 0, \text{ except } P_{4,4}(0) = 1
\end{aligned}
\qquad \text{[Eq. 2-7]}
$$

where $\kappa = k_b/k_r$ and the (m, 0) and (0, n) states correspond to the break condition of 2 aircraft.

The Red side wins the engagement if all the Blues are gone, that is, if the engagement is in a state (0, n), $r_{min} \leq n \leq N$. The Blue side wins if the engagement is in a state (m, 0), $b_{min} \leq m \leq M$.

Note that the first four equations in Equation 2-7 describe the evolution of the probabilities of transient states, whose probabilities will tend to zero at long times. The remaining four equations describe the variation of the absorbing boundary states, or outcome states. At long times, the system will be found in one of these states.

The equations for the transient states decouple from those of the outcome states. That is, the equations involve only the four transient-state probabilities. The evolution of the transient states can thus be determined independently of the outcome states.

The outcome states' evolution is, by contrast, completely determined by the transient states and the initial conditions. For example, given $P_{4,3}(t)$, the entire history of $P_{4,0}(t)$ follows from

$$\int_0^t \dot{P}_{4,0}(\tau)d\tau = P_{4,0}(t) - P_{4,0}(0) = P_{4,0}(t) = 4\kappa \int_0^t P_{4,3}(\tau)d\tau \qquad \text{[Eq. 2-8]}$$

where we have used the initial condition $P_{4,0}(0) = 0$.

Letting $t \to \infty$ in Equation 2-8, we see that the long-time limiting value of $P_{4,0}(t)$ is given by

$$\lim_{t \to \infty} P_{4,0}(t) = 4\kappa \int_0^\infty P_{4,3}(\tau)d\tau \quad . \qquad \text{[Eq. 2-9]}$$

We now show how to exploit characteristics of the engagement we are considering, shown by Figure 2-3 and reflected in properties of Equation 2-7, to allow calculation of long-time limiting probabilities of outcome states with a simple, finite iterative scheme. This engagement is an example of what have been called "pure death" processes: the states always transition to states with fewer participants. Reflecting this, Figure 2-3 is an "acyclic" graph. That is, the graph gives no path by which to return to a previously occupied state.

A consequence of the acyclic character of Figure 2-3 is found in the structure of Equation 2-7. Defining the vector x as (P_{44}, P_{43}, P_{34}, P_{33}), we see that the four equations for the transient states can be written as

$$\dot{x} = Ax \qquad \text{[Eq. 2-10]}$$

where the matrix A is given by

$$A = \begin{pmatrix} -(4\kappa+4) & 0 & 0 & 0 \\ 4k & -(4\kappa+3) & 0 & 0 \\ 4 & 0 & -(3\kappa+4) & 0 \\ 0 & 3 & 3\kappa & -(3\kappa+3) \end{pmatrix}. \qquad \text{[Eq. 2-11]}$$

The acyclic character of Figure 2-3 is reflected in the lower-triangular character of the matrix A. This lower-triangular feature allows calculation of the integrals from 0 to ∞ of each of the transient-state probabilities with a simple iterative scheme. Defining the vector $\hat{x}$ by

$$\hat{x} \equiv \int_0^\infty x(t)\,dt, \qquad \text{[Eq. 2-12]}$$

we see, on integrating Equation 2-10 from $t = 0$ to $t = \infty$ and remembering that $P_{4,4}(0) = 1$ and all other transient-state probabilities are zero at $t = 0$, that

$$-e_1 = A\hat{x}, \qquad \text{[Eq. 2-13]}$$

where e_1 is the vector (1, 0, 0, 0). Thus $\hat{x}$ is determined by the solution of a system of linear algebraic equations. Moreover, by virtue of the lower-triangular character of A, that system is easily solved with an obvious iterative scheme. On writing out Equation 2-13 in detail, we find

$$\begin{aligned} -1 &= -(4\kappa+4)\hat{x}_1 \\ 0 &= 4\kappa\hat{x}_1 - (4\kappa+3)\hat{x}_2 \\ 0 &= 4\hat{x}_1 - (3\kappa+4)\hat{x}_3 \\ 0 &= 3\hat{x}_2 + 3\hat{x}_3 - (3\kappa+3)\hat{x}_4 \end{aligned} \qquad . \qquad \text{[Eq. 2-14]}$$

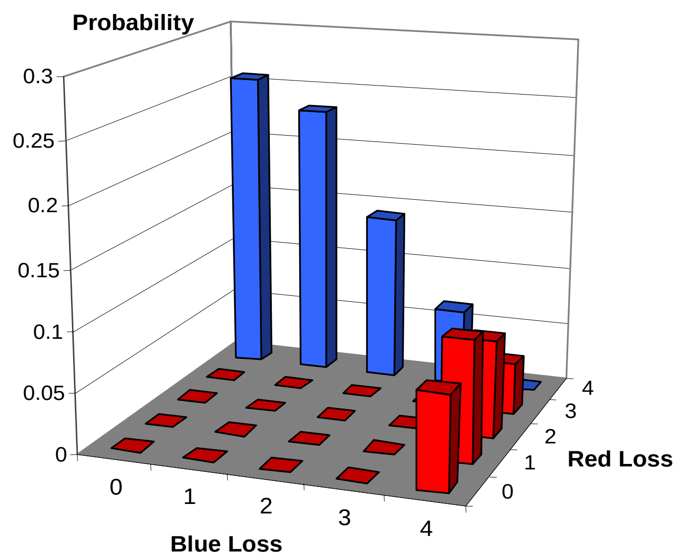
To solve those equations, one may obtain $\hat{x}_1$ from the first equations, then $\hat{x}_2$ and $\hat{x}_3$ follow from the second and third equations, and finally, $\hat{x}_4$ follows from the last equation. The lower-triangular character of the matrix A guarantees that the n^{th} equation in the system of Equation 2-14 involves $\hat{x}_n$ and does not involve $\hat{x}_j$ for any $j > n$. Another way to view the benefits of the lower-triangular character of the matrix A is to observe that this makes Equation 2-13 have the form of a system of equations after the forward course of Gaussian elimination has been performed, so that only the rapid back course remains to be done.

Why were we justified in integrating the $\hat{x}$ from 0 to ∞? Texts on differential equations[4] show that solutions of equations such as Equation 2-10 are always linear combinations of functions $\exp(\lambda_j t)$, where λ_j is an eigenvalue of the matrix A, when these eigenvalues are distinct. The eigenvalues of a lower-triangular matrix are the elements on the main diagonal. Inspection of Equation 2-11 shows that these elements are distinct, and negative. Thus solutions of Equation 2-7 decrease exponentially with time, and the integrals we use exist.

In Appendix A, we show that the approach to calculating long-time limiting probabilities illustrated here can be applied to many engagement models. We use this method, which from now on we will call "the method of Appendix A," frequently in the present SLAACM and in planned extensions to it.

For SLAACM, the key result from engagement calculations is the discrete bivariate loss distribution. This gives the loss statistics that are propagated day-by-day in a campaign. Figure 2-4 shows an example of the discrete bivariate loss distribution, for a 4 vs. 4 engagement in which both sides fight to annihilation (breakpoint = 0), and $\kappa = 1.46$. In the figure, blue bars indicate Blue win states (states with four Red losses), and red bars indicate Red win states (states with four Blue losses). The figure shows that, as should be the case when $\kappa > 1$, the Blues are likely to win the engagement and the total probability of the blue bars is distinctly larger than the total probability of the red bars. There is significant dispersion in both Red and Blue losses.

Figure 2-4. Bivariate Loss Distribution

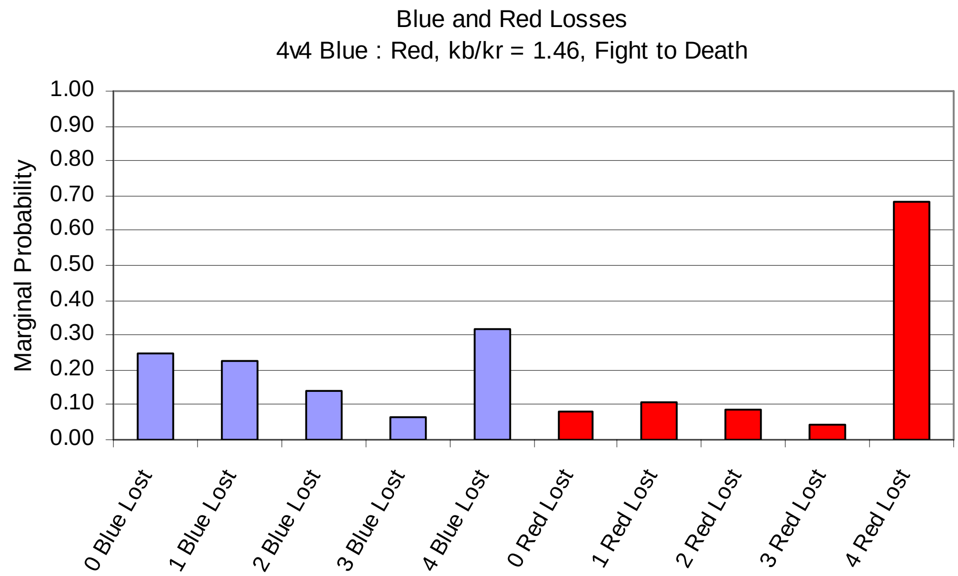


The marginal distributions of Blue and of Red losses and the total win probabilities are shown in Figure 2-5. The marginal distributions show that the marginal

[4] W.E. Boyce and R.C. DiPrima, *Elementary Differential Equations*, Third Edition (New York: Wiley, 1976), Sections 7-6 through 7-9.

probabilities are not necessarily monotone. They also show that, in this case, the probability of 4 losses is the largest Blue probability.

Figure 2-5. Marginal Loss Distributions with KRR=1.46

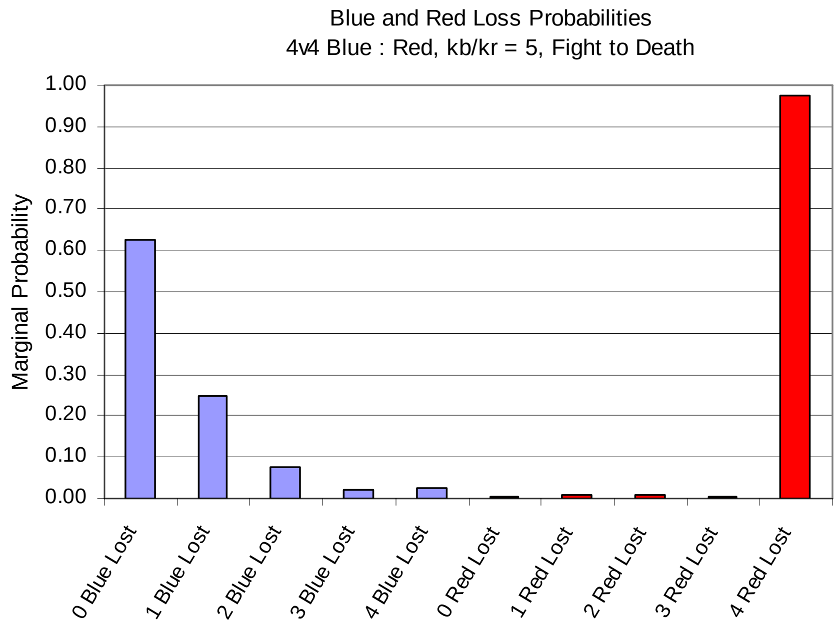


Such a "toe-up" at the greatest possible loss may seem counter-intuitive. In fact, that feature is certain to happen for kill-rate ratios κ sufficiently close to 1. To see this, consider a 4 vs. 4 engagement to annihilation, with $\kappa = 1$. The probability of a Blue win must equal the probability of a Red win, and these probabilities must both equal 0.5. The probability of a Red win is the probability of 4 Blue losses, so the probability of 4 Blue losses must be 0.5.

The sum of the marginal probabilities of Blue losses must be 1. Therefore, none of the marginal probabilities of 0, 1, 2, or 3 Blue losses can exceed 0.5. If, as in fact is the case, all the marginal probabilities of 0, 1, 2, and 3 Blue losses are positive, each of them must be less than 0.5, so that, in this case of equal strength forces ($\kappa = 1$), the marginal probability of 4 Blue losses must be the largest marginal loss probability.

The marginal loss probabilities are continuous functions of κ (they are in fact rational functions of κ, as may be seen by applying the method of Appendix A), so that, as κ increases from 1, at least for a while, the most probable Blue loss must be 4. Figure 2-6 shows that the Blue toe-up is negligible for $\kappa = 5$.

Figure 2-6. Marginal Loss Distribution



SCALABILITY AND COPING WITH LARGE-DIMENSIONED PROBLEMS

The examples illustrate that initial value problems (Equations 2-5 and 2-6) are tractable, and that their solutions give results that military planners may find useful. These examples also show why computational probabilistic models have not been widely used: Even moderate values of M and N generate a lot of equations. For example, an engagement of 4 Blues with 8 Reds, not at all uncommon, leads to a system of 44 differential equations. The more sophisticated models that we consider in Appendixes B and D can lead to systems of thousands of equations. Computational probabilistic models have significant scalability concerns.

Until fairly recently, it was not practical to deal with such large state spaces. Advances on two fronts have changed that, however. First, memory and speed of widely available PCs now match or exceed those of the supercomputers of only a decade or so ago. Second, recently available applications packages make it possible to treat the large-dimensioned probabilistic models and the many-variable integer programming problems, to which computational probabilistic models sometimes lead.

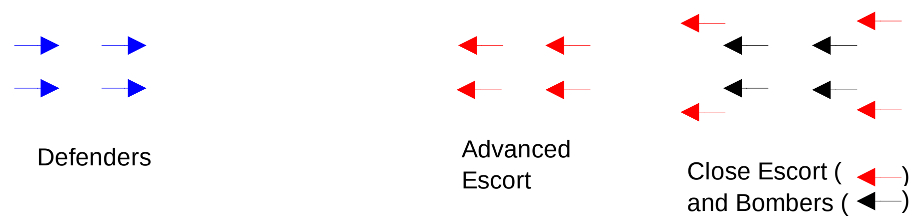
In the development of SLAACM, and for offline analyses, we have used Markov analysis tools developed by NASA for analysis of high-reliability electronics that greatly facilitate treating the large-dimensioned initial value problems to which computational probabilistic models often lead. Appendix B discusses how we have applied these tools to air combat analysis.

Composite Engagements

In the current version of SLAACM, engagements are always composite engagements of a particular kind, in which a flight of 4 defending fighters engages an attack package made up of 4 advanced escorts, 4 close escorts, and 4 bombers.

The defenders first engage the advanced escorts. If the defenders win that engagement, the surviving defenders engage the close escorts. If the defenders again prevail, defenders who survive the engagement with close escorts engage the bombers. Results of this final engagement determine the distribution of the number of bombers "leaking" through the defense. Figure 2-7 illustrates the opposing forces for this composite engagement.

Figure 2-7. Opposing Forces in SLAACM Composite Engagement



Sixteen final outcome or absorbing states are possible when both sides fight to annihilation. The outcome states may be described by the ordered quadruple (A, C, B, D), where A is the number of surviving Red advanced escorts, C the number of Red close escorts that survive, B the number of surviving Red bombers, and D the number of surviving Blue defenders. Table 2-1 lists the 16 possible absorbing quadruples for the SLAACM composite engagement. Table 2-2 shows the 64 total transition and absorbing states possible for the scenario.

Table 2-1. Absorbing States, SLAACM Composite Engagement

A Red LE	C Red CE	B Red B	D Blue
4	4	4	0
3	4	4	0
2	4	4	0
1	4	4	0
0	4	4	0
0	3	4	0
0	2	4	0
0	1	4	0
0	0	4	0
0	0	3	0
0	0	2	0

Table 2-1. Absorbing States,
SLAACM Composite Engagement

A Red LE	C Red CE	B Red B	D Blue
0	0	1	0
0	0	0	4
0	0	0	3
0	0	0	2
0	0	0	1

Table 2-2. Transition and Absorbing States,
SLAACM Composite Engagement

A Red LE	B Red CE	C Red B	D Blue
4	4	4	4
4	4	4	3
4	4	4	2
4	4	4	1
4	4	4	0
3	4	4	4
3	4	4	3
3	4	4	2
3	4	4	1
3	4	4	0
2	4	4	4
2	4	4	3
2	4	4	2
2	4	4	1
2	4	4	0
1	4	4	4
1	4	4	3
1	4	4	2
1	4	4	1
1	4	4	0
0	4	4	4
0	4	4	3
0	4	4	2
0	4	4	1
0	4	4	0
0	3	4	4

Table 2-2. Transition and Absorbing States, SLAACM Composite Engagement

A Red LE	B Red CE	C Red B	D Blue
0	3	4	3
0	3	4	2
0	3	4	1
0	3	4	0
0	2	4	4
0	2	4	3
0	2	4	2
0	2	4	1
0	2	4	0
0	1	4	4
0	1	4	3
0	1	4	2
0	1	4	1
0	1	4	0
0	0	4	4
0	0	4	3
0	0	4	2
0	0	4	1
0	0	4	0
0	0	3	4
0	0	3	3
0	0	3	2
0	0	3	1
0	0	3	0
0	0	2	4
0	0	2	3
0	0	2	2
0	0	2	1
0	0	2	0
0	0	1	4
0	0	1	3
0	0	1	2
0	0	1	1
0	0	1	0
0	0	0	4

Table 2-2. Transition and Absorbing States, SLAACM Composite Engagement

A Red LE	B Red CE	C Red B	D Blue
0	0	0	3
0	0	0	2
0	0	0	1

The first engagement is 4 defenders vs. 4 advanced escorts. Eight outcome states are possible from this engagement: four Red win states—(4, 4, 4, 0), (3, 4, 4, 0), (2, 4, 4, 0), and (1, 4, 4, 0)—in which the advanced escorts defeat the four defenders with 0, 1, 2, or 3 losses. Four Blue win outcomes are possible: (0, 4, 4, 4), (0, 4, 4, 3), (0, 4, 4, 2), and (0, 4, 4, 1), when the Blues defeat the advanced escorts with 0, 1, 2, or 3 losses. We use the method of Appendix A to determine the probabilities of these states. If the advanced escorts win this engagement, the composite engagement ends.

The Blue win states cause the engagement to continue into engagements between 1, 2, 3, or 4 defenders and the 4 close escorts. Four possible outcome states are Red win states: (0, 4, 4, 0), (0, 3, 4, 0), (0, 2, 4, 0), and (0, 1, 4, 0), corresponding to close escort losses of 0, 1, 2, or 3 aircraft.

The engagement between 1 defender and the 4 close escorts has one Blue win state: (0, 0, 4, 1). The engagement between 2 defenders and 4 close escorts has two Blue win states: (0, 0, 4, 1) and (0, 0, 4, 2). The engagement between 3 defenders and 4 close escorts has three Blue win states—(0, 0, 4, 1), (0, 0, 4, 2), and (0, 0, 4, 3)—and the engagement between 4 defenders and 4 close escorts has four Blue win states: (0, 0, 4, 1), (0, 0, 4, 2), (0, 0, 4, 3), and (0, 0, 4, 4). Again, we use the method of Appendix A to evaluate the probabilities of all these outcomes.

Thus the engagements with close escorts lead to four Red win states and to four cases of engagements between defenders and bombers: 1 vs. 4, 2 vs. 4, 3 vs. 4, and 4 vs. 4. These engagements lead to the Red win states (0, 0, 1, 0), (0, 0, 2, 0), (0, 0, 3, 0), and (0, 0, 4, 0) and to the Blue win states (0, 0, 0, 1), (0, 0, 0, 2), (0, 0, 0, 3), and (0, 0, 0, 4). Using the method of Appendix A to evaluate the probabilities of the outcome states in the defenders versus bomber engagements completes determination of the probabilities of all 16 outcome states in Table 2-1.

SLAACM contains individual kill-rate ratios for each Blue-Red aircraft pair in the Blue and Red inventories. SLAACM uses these to calculate probabilities for each of the 16 possible outcomes for all the feasible 4 vs. 4+4+4 combinations. In a campaign, Red and Blue use the resulting probabilities to construct optimal attack packages and optimal defenses.

EXTENSIONS TO ENGAGEMENT MODELS

We have now reviewed the basic engagement model, and the composite engagement model, presently used in SLAACM. The same methods used here can also be used to extend SLAACM by using more sophisticated engagement models, such as the ones described in Appendix D. Given more data than the presently available loss ratios, for example, we could introduce the more realistic two-phase kill engagement model of Appendix D. When breaking radar lock is not considered, the long-time limiting values of outcome probabilities for this model are determined by three non-dimensional ratios:

- Ratio of the mean time for lock-on by the Blue aircraft to the mean time for lock-on by the Red aircraft
- Ratio of the mean time for the Blue aircraft to make a kill after lock-on to the mean time for lock-on by the Red aircraft
- Ratio of the mean time for a Red aircraft to make a kill after lock-on to the mean time for lock-on by the Reds.

When breaking radar lock is considered, two more ratios are added.

Given at least three outcome data, we could infer the non-dimensional ratios needed for the two-phase kill model without lock-breaking. With at least five outcome data, we could infer the ratios needed for the two-phase kill model with lock-breaking.

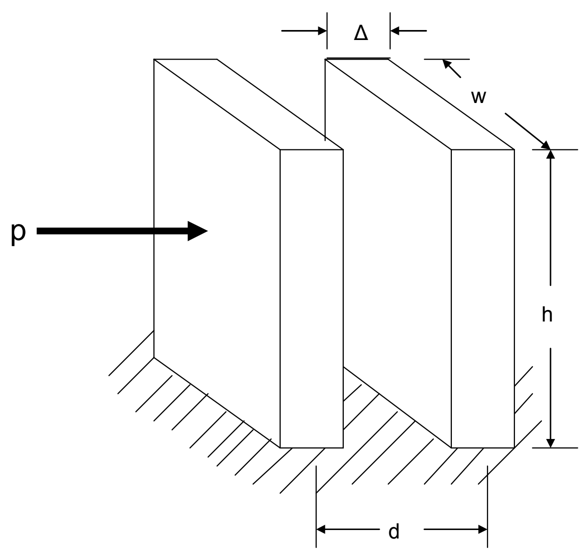
SLAACM's composite engagements also can be extended, for example, by including the effects of SAM and AAA defenses (defensive counter air effects are already included). Engagement models of the kind described above for air-to-air engagements can also be made for suppression of enemy air defense (SEAD) actions; those engagements could also be included in SLAACM's composite engagements.

Chapter 3
How SLAACM Works: Bomber Effectiveness Parameter

The basic campaign modeled in SLAACM is a Red bomber attack against Blue defenders. Red's bombers have different levels of effectiveness, based on the attack package. The offensive effectiveness of a bomber depends on two factors: the payload of bombs carried, and the accuracy with which those weapons can be delivered to targets. We developed a measure of bomber effectiveness based on these factors.

We considered first the relation between blast overpressure p and the properties of a structure designed to resist that overpressure. We used the simple model of reinforcing structure shown in Figure 3-1.

Figure 3-1. Reinforcing Structure Model



In this simple model, two columns reinforce the structure. Their thicknesses are Δ, their heights are h, and the spacing between their centers is d. Their dimension into the page is w.

The reinforcing columns act as cantilevers against lateral loads. A pressure p acting on the left face of this simple structure will induce axial loads $\frac{pwh^2}{2d}$ in each column, tension in the left and compression in the right. The pressure will also

induce shear loads $\frac{pwh}{2}$ in each column. Since both loads are proportional to p, the maximum stress induced in the columns will also be proportional to p. If the columns' dimensions are adjusted so that a material of given maximum allowable stress can support the loads induced by the overpressure, the area of the columns' bases $w\Delta$ will be proportional to the maximum stress and, therefore, also to p.

Thus, the volume of the reinforcing structure, $2w\Delta h$, will be proportional to p. Assuming that the cost of the reinforcing structure is proportional to its volume leads to the conclusion that the cost of reinforcement will be proportional to overpressure p.

Now let us consider how blast overpressure varies with explosive yield E. Dynamic overpressure p is proportional to the square of the air velocity v immediately behind the blast wave produced by the explosion. Dimensional analysis shows how v varies with E.[1] In addition to E, the parameters affecting v are the undisturbed air density ρ and the time t required for the blast wave to reach the locations of interest.

According to the principles of dimensional analysis, v can be expressed as the product of a term $E^{\alpha}\rho^{\beta}t^{\gamma}$ and a dimensionless function of all possible dimensionless combinations of E, ρ, and t. If $E^{\alpha}\rho^{\beta}t^{\gamma}$ is equal to a velocity, that term must have the dimension L/T, where L represents dimension length and T dimension time. Now, energy E has dimension ML^2/T^2, where M denotes dimension mass; ρ has dimension M/L^3, and, of course, t has dimension T. Thus

$$\left(\frac{ML^2}{T^2}\right)^{\alpha}\left(\frac{M}{L^3}\right)^{\beta}T^{\gamma} = \frac{L}{T}. \qquad \text{[Eq. 3-1]}$$

Equation 3-1 implies that

$$\begin{aligned} \alpha + \beta &= 0 \\ 2\alpha - 3\beta &= 1 \\ -2\alpha + \gamma &= -1 \end{aligned} \quad . \qquad \text{[Eq. 3-2]}$$

The unique solution to Equation 3-2 is $\alpha = 1/5, \beta = -1/5, \gamma = -3/5.$ Similar arguments to those leading to Equations 3-1 and 3-2 show that there are no non-dimensional combinations of E, ρ, and t, other than a constant. Therefore,

$$v \propto \left(\frac{E}{\rho}\right)^{\frac{1}{5}} t^{-\frac{3}{5}} \qquad \text{[Eq. 3-3]}$$

---

[1] L. Sedov, *Similarity and Dimensional Methods in Mechanics* (New York: Academic Press, 1959).

and the dynamic overpressure p will satisfy

$$p \propto \rho v^2 \propto E^{\frac{2}{5}} \rho^{\frac{3}{5}} t^{-\frac{6}{5}}. \qquad \text{[Eq. 3-4]}$$

Similar dimensional considerations lead to the conclusion that the radius r of the blast wave satisfies the proportionality

$$r \propto \left(\frac{E}{\rho}\right)^{\frac{1}{5}} t^{\frac{2}{5}}. \qquad \text{[Eq. 3-5]}$$

From proportionalities (Equations 3-4 and 3-5), it follows that the radius r_L at which the overpressure falls to p_L satisfies

$$r_L \propto \left(\frac{E}{\rho}\right)^{\frac{1}{3}} p_L^{-\frac{1}{3}}. \qquad \text{[Eq. 3-6]}$$

Thus, the lethal radius of a blast wave increases as the cube root of the blast's energy and decreases as the cube root of the lethal overpressure.

Since in our model the value of a structure is proportional to the overpressure required to destroy it, proportionality (Equation 3-6) implies that the lethal radius of a blast with energy E varies directly with the cube root of E, and inversely with the cube root of the value of the structures it affects.[2]

We now use the information about the variation of lethal radius with explosion energy and structure value to see how bombs' single shot kill probabilities (sspk's) vary with explosion energy and value of structures affected.

The sspk of a bomb with lethal radius r_L, delivered with circular error probable (CEP), is

$$\text{sspk} = 1 - \left(\frac{1}{2}\right)^{\frac{r_L^2}{\text{CEP}^2}}. \qquad \text{[Eq. 3-7]}$$

Introducing the variation of r_L with explosive energy E and value V obtained above, we have

$$\text{sspk} = 1 - \left(\frac{1}{2}\right)^{\frac{kE^{2/3}V^{-2/3}}{\text{CEP}^2}}, \qquad \text{[Eq. 3-8]}$$

[2] These conclusions, obtained with simple dimensional analyses, agree with those from the exact Taylor-Sedov solution [Sedov, l. c. ante] for values immediately behind the shock.

where k is a constant.

Now,

$$kE^{2/3}V^{-2/3} = kE_{ref}^{2/3}V_{ref}^{-2/3}\left(\frac{E}{E_{ref}}\right)^{\frac{2}{3}}\left(\frac{V_{ref}}{V}\right)^{\frac{2}{3}}\left(\frac{CEP_{ref}}{CEP}\right)^{2}. \qquad \text{[Eq. 3-9]}$$

Defining the parameter k_1 by

$$k_1 \equiv kE_{ref}^{2/3}V_{ref}^{-2/3}, \qquad \text{[Eq. 3-10]}$$

we have

$$sspk = 1-\left(\frac{1}{2}\right)^{k_1\left(\frac{E}{E_{ref}}\right)^{\frac{2}{3}}\left(\frac{V_{ref}}{V}\right)^{\frac{2}{3}}\left(\frac{CEP_{ref}}{CEP}\right)^{2}}. \qquad \text{[Eq. 3-11]}$$

Let E_{ref} be the energy of a 500 lb bomb. Let V_{ref} be the value of a target hardened to withstand a 500 lb bomb. We will consider 500 lb, 1,000 lb, and 2,000 lb bombs and assume that bombs are always matched to targets, so that a bomber on a mission against 500 lb targets carries 500 lb bombs, and one on a mission against 1,000 lb targets carries 1,000 lb bombs, and that one going against 2,000 lb targets carries 2,000 lb bombs.

With those assumptions, the product of the term involving E and the term involving V in Equation 3-11 is always 1, and

$$sspk = 1-\left(\frac{1}{2}\right)^{k_1\left(\frac{CEP_{ref}}{CEP}\right)^{2}}. \qquad \text{[Eq. 3-12]}$$

We arbitrarily set the constant k_1 to 1; we only want a systematic way to compute the value to Red of a bomber with given payload P that is capable of delivering bombs with relative accuracy CEP/CEP_{ref}.

With this assumption, the value of the expected number of targets killed by such a bomber carrying 500 lb bombs is

$$V_1 = \left\lfloor \frac{P}{500} \right\rfloor sspk_0 V_{500}, \qquad \text{[Eq. 3-13]}$$

where

$$\mathrm{sspk}_0 = 1 - \left(\frac{1}{2}\right)^{\left(\frac{\mathrm{CEP_{ref}}}{\mathrm{CEP}}\right)^2} \qquad \text{[Eq. 3-14]}$$

and where V_{500} is the value associated with a target hardened to withstand 500 lb bombs.

When the bomber is loaded with 1,000 lb bombs, the value of the expected number of targets killed is

$$V_2 = \left\lfloor \frac{P}{1000} \right\rfloor \mathrm{sspk}_0 (2)^{2/5} V_{500}\,, \qquad \text{[Eq. 3-15]}$$

since the value of a target goes as the 2/5 power of the explosive yield required to destroy it.

Finally, if the bomber is loaded with 2,000 lb bombs, the value of the expected number of targets killed is

$$V_2 = \left\lfloor \frac{P}{2000} \right\rfloor \mathrm{sspk}_0 (4)^{2/5} V_{500}\,. \qquad \text{[Eq. 3-16]}$$

Using this information, SLAACM assigns the average of V_1, V_2, and V_3, divided by V_{500}, as the value of the bomber.

SLAACM currently assigns CEP/$\mathrm{CEP_{ref}}$ ratio to 1 if the bomber cannot deliver precision-guided (smart) bombs. With payloads of smart bombs the CEP/$\mathrm{CEP_{ref}}$ will be less than 1 and the ($\mathrm{CEP_{ref}}$/CEP) term in the sspk_0 equation will be greater than 1. If specific values of CEP are available, they can be used in the calculation of bomber value.

Although bombers are considered to have poor defensive capability, they do have kill-rate ratios assigned and their defensive capability is considered along with their offensive capability in the payoff calculations (discussed in the next chapter).

Chapter 4
How SLAACM Works: Campaign Logic and Optimization

This chapter discusses the Red and Blue payoff logic and optimization algorithms used in SLAACM.

SPECIFYING RED AND BLUE PAYOFF FUNCTIONS

SLAACM determines optimal Red attacks and optimal Blue responses using payoff functions for each side. This section describes those calculations.

Red's Payoff Function

Red values an attack package by considering the benefit of getting bombers through and of downing Blue aircraft. Red may also consider penalties for losing its own aircraft, assigning different values to losing a fighter and to losing a bomber. Specifically, a package's value to Red is determined in this way: A Red package is defined by the triple (a, c, b), where a is the advanced escort type, c the close escort type, and b the bomber type (the bombers may be fighter/bomber aircraft). The outcome statistics for an engagement between the package and the "planning Blue" aircraft (the user identifies a Blue type for Red's planning) are determined, and, from these, the expected number B_s of surviving bombers, the expected number RF_k of lost fighters, and the expected number D_k of downed Blues are computed. Then V_R, Red's value for the package, is given by

$$V_R = c_1 B_s + c_2 D_k - c_3 RF_k - c_4 (4 - B_s) . \qquad \text{[Eq. 4-1]}$$

The positive parameter c_1 is currently set at the value of a Red bomber, computed by the method described in Chapter 3. The value of c_2 is set at 20, reflecting the fact that Red believes Blue to be loss averse and likely to abandon the campaign if their side incurs many losses. Currently, c_3 and c_4 are both set at zero. When the Blue aircraft are significantly stronger than the Red aircraft (values of κ significantly greater than 1), even values of c_3 and c_4 that are small compared to 1 may cause the Reds not to dispatch any packages at all.

Blue's Payoff Function

Blue values engaging a given Red package by considering the benefit of downing Red fighters and bombers, and the penalty of losing its own aircraft. Specifically, the value to Blue of engaging a Red package with fighters of a given type is cal-

culated by computing the outcome statistics for that composite engagement (described in Chapter 3) and, from them, the expected number B_k of bombers killed, the expected number RF_k of Red fighters killed, and the expected number D_k of defenders killed. Then the value V_B of the engagement to Blue is evaluated as

$$V_B = d_1 B_k - d_2 D_k + d_3 RF_k . \qquad \text{[Eq. 4-2]}$$

Currently, d_1 is set to Red's value for the bomber type in the package, d_2 is set to 20, and d_3 is set to 1.

Values of c_1, c_2, c_3, c_4, d_1, d_2, and d_3 are not currently user inputs to SLAACM; they may, however, readily be changed by simple modifications to the code.

Determining Attacks and Defenses

A key part of SLAACM's operation is determining the engagements that occur each day. This is done by finding optimal sets of attack packages dispatched by Red and by determining Blue's response in two ways: the response of smart Blues who know the makeup of Red's packages before intercepting them and make optimal defenses, and the response of other not-smart Blues who encounter Red packages randomly. This section explains these calculations.

Determining Red's Attack Packages

As the aggressor, Red always plans its attacks in advance. This leads to an integer programming (IP) problem, which SLAACM treats in one of two ways: a classical "greedy" heuristic or an exact solution using the commercial general optimization package LINGO™. This section describes those calculations.

Red's Integer Programming Problem

Each day, Red has a set of fighter, bomber, and fighter/bomber aircraft to deploy. In SLAACM, Red chooses the set of attack packages to maximize the total of the expected Red payoff of the packages dispatched (evaluating payoff as described above in "Red's Payoff Function"), constrained by the available forces.

Here we formalize the IP problems performed in the LINGO™ model within SLAACM. The main idea behind using an IP tool is to solve the following optimization problem: Red wishes to come with optimal attack packages based on a determined payoff function (this is the first optimization). Then, the Blue aircraft that are sophisticated enough to have a priori knowledge of what Red attack packages are coming may optimize their defending packages accordingly (the second optimization).

A Red attack package consists of 4 advanced escorts, 4 close-in escorts, and 4 bombers. Since some fighters may be bombers, we could have a set of possible attack packages T, where, given R types of Red fighters and bombers, we could

have $|T| = R^3$. In reality, however, most mathematically feasible combinations of packages are unrealistic for warfare (one would not, for example, dispatch a package of heavy bombers escorting fighters). Since the number of reasonable attack packages is much smaller than the complete enumeration, we assume those undesirable packages are removed in advance of the optimization; then, we limit ourselves to those remaining, reasonable possibilities in the integer program. The degree to which one wishes to limit the number of potential packages depends primarily on the size of the integer program, which is a function of the size of R.

We define a set of attack packages for Red as $I \subseteq T$. We define variable r_{ij} as the number of Red aircraft of type j used in attack package i. To calculate its payoff, Red plans its attack assuming each package will be confronted by one specific Blue defender type, usually the most numerous of the Blues. Based on the payoff functions described previously, each attack package for Red has a certain payoff, denoted as p_i. We denote the number of Red aircraft of type j as n_j. At this point, Red can solve for its optimal attack strategy, denoted by the following integer optimization problem:

$$\begin{aligned}
&\max \sum_{i=1}^{I} p_i x_i \\
&\text{s.t.} \ \sum_{i=1}^{I} r_{ij} x_i \le n_j \quad \forall j \in J. \\
&x_i \in \{0,1,2,\ldots\}
\end{aligned}$$

We solve for the values of x_i's, which represent the optimal number of packages of type i sent. This problem is a variant of the classic knapsack problem. The solution to this integer programming problem provides the optimal set of attack packages dispatched by Red.

HEURISTIC SOLUTION

The exact integer programming problem can be solved using an IP software tool such as LINGO. In SLAACM, the user can specify that the exact solution technique to be used, provided the user's PC has a LINGO license. Although utilizing an optimization package is the ideal circumstance—it can guarantee optimal solutions if run to completion—many analysts will not have LINGO available on their PCs. For this reason, SLAACM includes a greedy heuristic using Red and Blue pay-off functions that attempts to approximate the optimum solution. The heuristic should yield a near-optimum solution when the Red and Blue aircraft types and numbers are limited. However, as the choices grow, and particularly when Red has options of using aircraft as fighters or fighter/bombers, the packages selected by the heuristic will deviate significantly from an IP solution. This difference will be compounded when Blues are smart and also optimize. (When Blue

aircraft are dominant, the different package choices may produce only small differences in the outcome statistics.)[1]

The greedy heuristic does what might be considered the "obvious" answer. It is considered greedy because it takes, without more holistic considerations, packages with the largest payoffs first. This approach can be suboptimal because it does not consider (or mathematically eliminate from consideration) all combinations of package types.

As we noted earlier, the payoff to Red for attack package i is denoted as p_i. The heuristic first orders the packages from greatest to least. Without loss of generality, we denote the largest payoff as package p_1, the second largest as, p_2, etc. Thus, we have $p_1 \leq p_2 \leq ... \leq p_I$. Again, we define n_j as the number of Red aircraft of type j. The heuristic can then be described by the following procedure.

Step 0: Set $\tilde{n}_j = n_j$ for all j.

Initialize $x_i = 0$ for all i.
Set $i = 1$.

Step 1: If $\tilde{n}_j - 1 \geq 0$ for all j such that $r_{ij} > 0$, then {

$\tilde{n}_j = \tilde{n}_j - 1$ for all j such that $r_{ij} > 0$
$x_i = x_i + 1$

Go to Step 1

} Else {

Go to Step 2 }

Step 2: $i = i + 1$.

If $p_i \leq 0$, quit.
If $i = I + 1$, quit.
Else, Go to Step 1.

In essence, this heuristic keeps selecting the package with the greatest payoff until it exhausts one of the package's aircraft types (i.e., "hits a constraint"). It then looks for the package with the next greatest payoff (assuming all its aircraft types are still available). It keeps selecting that package until one of its aircraft types is exhausted, and so forth. It continues this procedure until there are no more packages with positive

[1] For further information on integer programming and greedy heuristics, the interested reader should consider one of the many textbooks on integer programming and optimization, such as L. Wolsey and G. Nemhauser, *Integer and Combinatorial Optimization* (New York: Wiley, 1988).

payoffs for which the advanced escort, close-in escort, or bomber aircraft types have not been exhausted.

It is important to reemphasize that this approach will not necessarily yield an optimal solution. Indeed, for sample campaigns modeled in SLAACM, the above heuristic typically does *not* yield the same solution for Red as the IP solution solved to optimality in LINGO. For the purposes of SLAACM, it appears that it is rare for the heuristic solutions to differ by more than a few percent from optimality. Nevertheless, theoretical bounds on such a heuristic for a *general* IP problem can be rather far from optimality.

EXACT SOLUTION

Red's IP problem may be solved exactly using an integer programming solver. LINGO is a commercially available general linear, nonlinear, and integer programming solver tool.[2] The version embedded in SLAACM is Extended LINGO™ 9.0, which does not limit the number of variables (integer or otherwise) or constraints. Of course, for a large enough problem, even sophisticated IP solvers will have trouble producing the optimal solution in reasonable amounts of time. The example problems run in SLAACM, but rarely take more than a couple of seconds on a modern desktop PC.

The IP model below is written in LINGO's modeling language and implemented in SLAACM for Red's optimization. The data and variables are defined elsewhere in SLAACM. For the sake of clarity, we omit portions of the data, variable definitions, and input/output declarations.

```
MODEL:
SETS:
RPACKAGE/1.rposize/: rpo, X;
REDAC/1.rac/: rob;
RMATRIX(RPACKAGE,REDAC): rcnst;
ENDSETS
! INTEGER PROGRAMMING PROBLEM;
MAX = VALUE;
VALUE = @SUM(RPACKAGE: rpo*X);
@FOR( REDAC(J):
@SUM(RPACKAGE(I): rcnst(I,J)*X(I)) <= rob(J) );

@SUM(RPACKAGE(I): X(I)) <= (redLim/12);
@FOR( RPACKAGE: @GIN(X));
END
```

[2] Lindo Systems, Inc., http://www.lindo.com/.

Determining Blue's Response

Certain Blue aircraft are identified by the user as smart. They are assumed to know the composition of Red's attack packages before intercepting them and to coordinate their defense to maximize its total value (value computations are described above in the section "Blue's Payoff Function"). The remaining Blue aircraft encounter Red packages randomly. The following subsections describe how SLAACM determines the set of engagements made by smart Blues and by the remaining Blue aircraft.

Blue's Integer Programming Problem

Given the set of Red attack packages, the smart Blues determine which types to engage by maximizing the total Blue payoff, subject to constraints imposed by their order of battle. As stated previously, Blue responds with optimal defenses of Red's attack if a given aircraft is smart enough to have a priori knowledge of Red's attack package composition. We assume Red has obtained its solution before Blue calculates its optimal response. We denote the optimal solution for Red—that is, $(a_1, a_2, ..., a_I) = (x_1^*, x_2^*, ..., x_I^*)$ —as constraints for Blue's optimization. In other words, a_i is the number of Red attack packages of type i that Red elected to send.

Blue has its own payoff function for each type of potential Red attack package. We denote $\hat{p}_{ik}$ as the payoff to Blue for intercepting a Red package i with the Blue 4-ship package of aircraft type k. Again, the components of the payoff function have been described previously; they include a loss-aversion factor for Blue, along with positive payout for expected Red kills and bomb damage prevented. We denote the set of Blue aircraft types with a priori knowledge of Red's attack packages as $K' \subseteq K$. We denote the number of Blue aircraft of type $k \in K'$ as m_k. We then formulate and solve the following IP problem for Blue. In short, we want to solve for the optimal number of Blue 4-ship package of type k that intercepts Red attack package i, denoted by y_{ik}.

$$\max \sum_{i=1}^{I} \sum_{j=1}^{K'} \hat{p}_{ik} y_{ik}$$

$$\text{s.t.} \sum_{i=1}^{I} 4y_{ik} \le m_k \quad \forall k \in K'$$

$$\sum_{k=1}^{K'} y_{ik} \le a_i \quad \forall i \in I$$

$$y_{ik} \in \{0,1,2,...\}. \qquad \text{[Eq. 4-3]}$$

The IP model written in LINGO's modeling language is embedded into SLAACM. LINGO solves the IP problem and returns the solution to variables defined in SLAACM. At this point, the remaining not-smart Blue aircraft, denoted by $K'' \subseteq K$, where $K' \cup K'' = K$ and $K' \cap K'' = \emptyset$, engage the Red packages that were not selected by the smart Blues. That is, the Blue aircraft without a priori knowledge engage randomly the Red packages that remain after the above IP problem is solved, i.e., those Blue aircraft engage a set of attack packages $(a'_1, a'_2, ..., a'_I)$, where $a'_i \equiv \max\left(a_i - \sum_{k=1}^{K} y_{ik}, 0\right)$. Those encounters are described in a later section.

Heuristic Solution

As stated previously, SLAACM allows the user to specify whether to use LINGO to solve the IP formulation or to use a greedy heuristic. If SLAACM uses the heuristic (or solves the IP problem in LINGO), it must do so for both Red's and Blue's optimization.

The greedy heuristic for Blue works similarly to the heuristic for Red. Following the notation from the IP formulation, we denote $\hat{p}_{ik}$ as the payoff to Blue for intercepting a Red package i with the Blue 4-ship package of aircraft type k. We then order the $\hat{p}_{ik}$ from greatest to least over all i and k. We denote that new set of payoffs as $\hat{p}_1 \geq \tilde{p}_2 \geq ... \geq \tilde{p}_B$, and note their associated i and k.

It is worth noting that Blues that are not members of the local air defense (LAD) are presumed to be loss averse, and it is not uncommon for some $\hat{p}_{ik} < 0$, i.e., intercepting certain Red's attack packages may have negative payoffs for Blue. This result occurs because loss-averse Blues suffer a large penalty for their expected losses. Neither the heuristic nor the LINGO model uses any aircraft whose payoff values result in $\hat{p}_{ik} \leq 0$, since selecting those combinations would of course reduce Blue's objective function.

Again, we denote the number of Red packages of type i sent as a_i. As before, we denote the number of Blue aircraft of type $k \in K'$ as m_k. The set of decision variables (number of Blue 4-ship packages of type k that intercept Red attack package i) is denoted by y_{ik}. As with Red, we wish to solve a multidimensional knapsack problem.

We can then describe the optimization heuristic we employ by the following algorithm:

Step 0: Set $\tilde{m}_k = m_k$ for all k.

Set $\tilde{a}_i = a_i$ for all i.

Initialize $y_{ik} = 0$ for all i, k.

Begin with $b = 1$.

Step 1: Let $\hat{i}, \hat{k}$ be the values for $\hat{p}_{\hat{i}\hat{k}}$ such that $\hat{p}_{\hat{i}\hat{k}} = \hat{p}_b$

If $\tilde{m}_k - 4 \geq 0$ and $\tilde{a}_i > 0$, Then {

$\tilde{m}_k = \tilde{m}_k - 4$

$\tilde{a}_i = \tilde{a}_i - 1$

$y_{ik} = y_{ik} + 1$

Go to Step 1

} Else {

Go to Step 2}

Step 2: $b = b + 1$.

If $p_b \leq 0$, quit.

If $b = B + 1$, quit.

Else, Go to Step 1.

In essence, this heuristic finds the Blue k vs. Red i solution with the highest payoff to Blue; notes the associated Red and Blue package types (i.e., i and k, respectively); checks to see if there are 4 Blue aircraft of type k available; checks to see if Red sent any packages of type i; and chooses those Blue aircraft to send after that Red package. It sends the lesser of the number of Blue aircraft of that type still available or the number of Red packages of that type that were sent. Then, the heuristic looks for the next highest payoff and repeats the above process until all smart Blue aircraft with positive payoffs are assigned or until there are no more Red packages to assign a smart Blue 4-ship package.

As this process is similar to the one for the Red heuristic, it needs to be stated again that this method does not guarantee optimality for Blue, because it does not consider the optimization of the entire Blue fleet over all possibilities, but rather looks for a greedy local solution. That said, both Blue's and Red's heuristics are computationally easy to implement. Indeed, without the ability to solve moderately complex integer programs, these techniques can form good approximations. However, neither heuristic is able to provide tight bounds on its performance, at least for a general knapsack problem. Therefore, the preferred method, given the

choice of the two methods, is to solve both optimizations exactly using an IP solver,

Exact Solution

Blue's IP problem may be solved exactly using LINGO. In general, Blue's optimization is a little more complex than Red's, because it requires assigning Blue's defending packages to Red's attack packages. Red simply creates optimized attack packages, so Blue is effectively optimizing over two indices (which packages should be sent, and whom they should intercept). In most cases, however, the number of variables in Blue's optimization is not larger than the number of decision variables in Red's optimization, because Red sends only a small fraction of its potential package types. In both cases, LINGO obtains the optimal solution to the IP problem in a few seconds on a standard, modern desktop PC for most sample problems in SLAACM.

The IP model written below in LINGO's modeling language solves Blue's optimization problem. It is similar in syntax to Red's LINGO model. Again, the data and the variable names are defined elsewhere in SLAACM. For the sake of clarity, we omit most data and variable definitions, as well as input/output calls.

```
MODEL:
SETS:
PPACKAGE/1.bsize/: sboob;
REDAC/1.rsize/: rsent;
PMATRIX(REDAC, PPACKAGE): bpo, X;
ENDSETS

! INTEGER PROGRAMMING PROBLEM;
! Objective Function;
MAX = VALUE;
VALUE = @SUM(PMATRIX: bpo*X);
! subject to;
@FOR( PPACKAGE(J): @SUM(REDAC(I): X(I,J)*4) <= sboob(J) );
@FOR( REDAC(I): @SUM(PPACKAGE(J): X(I,J)) <= rsent(I) );
@FOR(PMATRIX(I,J): @GIN(X(I,J)));
END
```

Random Engagements by "Not-Smart" Blue Aircraft

The optimal engagements made by smart Blue aircraft may leave some Red packages unengaged. For determining the packages that need to be intercepted by not smart Blues, we assume, optimistically for Blue, that a Red package intercepted by a smart Blue flight does not continue the engagement. This assumption is rea-

sonable when the smart Blues are much stronger than any of Red's fighters and bombers.

Other SLAACM versions have considered the opposite polar case, which may be implemented in SLAACM with simple reprogramming, where the Red forces that survive the smart Blue defense reorganize into optimal attack packages and continue the assault.

Blue aircraft not identified as smart encounter Red packages randomly. They do not conduct optimized defensive encounters. There is still an element of battle management for these aircraft, however, and that is the efficiency with which they can locate and engage the Red attack packages.

With perfect battle management, if the number of defending flights is at least as large as the number of attack packages, every attack package would be intercepted. The intercepting aircraft types would be random, however, rather than optimal. If the number of attack packages is larger than the number of defending flights, every defending flight would engage an attack package.

With less-than-perfect battle management, not every attack package would be intercepted in the former case, and not every defending flight would be engaged in the latter case. SLAACM's battle management feature allows the user to enter a "goodness" parameter that characterizes the effectiveness of battle management for Blue aircraft that are not smart. The following paragraphs explain how that parameter affects SLAACM's calculations.

Let K distinct types of defender flights deal with an attack by J distinct types of attack packages. The defenders are assumed not to know the makeup of individual attack packages before interception, so that the type of attack package engaged by a given defending flight is the result of random selection from the set of attack packages.

Let m_i be the number of defender flights of type i, and let n_j be the number of attack packages of type j. Then the total number of attack packages, N, and the total number of defending flights, M, are given respectively by

$$N = \sum_{1}^{J} n_j ; \quad M = \sum_{1}^{K} m_i \,. \qquad \text{[Eq. 4-4]}$$

First we consider the case of perfect battle management. If $M \geq N$, then every attack package will be intercepted. Not all defending flights engage; in this simple analysis, we assume that the fraction N/M of each defending flight type engages.[3]

[3] We recognize that this simple analysis treats a complex combinatoric problem crudely.

Then, a measure of $\overline{E}_{ij}$, the central tendency of the number E_{ij} of m_i vs. n_j engagements, is

$$\overline{E}_{ij} = \frac{N}{M} m_i \frac{n_j}{N} = \frac{m_i n_j}{M}. \qquad \text{[Eq. 4-5]}$$

For this simple analysis, we take $\overline{E}_{ij}$ to be the number of m_i vs. n_j engagements. Summing $\overline{E}_{ij}$ over j shows that the fraction of each defending flight type engaged is N/M, as it should be; summing over i shows that all attacking flights of each type are engaged.

When M < N, every defending flight engages, but not all attack packages can be engaged. The estimate for $\overline{E}_{ij}$ analogous to the one given in Equation 4-5 is

$$\overline{E}_{ij} = m_i \frac{n_j}{N} = \frac{m_i n_j}{N}. \qquad \text{[Eq. 4-6]}$$

Summing this estimate for $\overline{E}_{ij}$ over i shows that the fraction M/N of each attack package type is engaged; summing over j shows that every defending flight of each type is engaged.

The Battle Management Efficiency Factor

In some cases, M and N are both $O(10^2)$. For such large M and N, providing this "perfect" defender's battle management could exceed the capabilities of available systems. We account for the limitations of the defender's battle management with a simple adaptation of the "perfect" case. We make the adaptation by introducing "ghost" attack packages or defender flights. A defender flight that engages a ghost attack package does not actually engage; an attack package engaged by a ghost defender flight is not actually intercepted.

When $M \geq N$, we introduce m_g defending flights of type "ghost," and proceed as in the above analysis (where the defender has perfect battle management). This gives us a new estimate for $\overline{E}_{ij}$:

$$\overline{E}_{ij} = \frac{m_i n_j}{M + m_g} = \frac{m_i n_j}{M} b_1, \qquad \text{[Eq. 4-7]}$$

where

$$b_1 \equiv \frac{M}{M + m_g}. \qquad \text{[Eq. 4-8]}$$

When $M < N$, we introduce n_g attack packages of type "ghost," and find

$$\overline{E}_{ij} = \frac{m_i n_j}{N + n_g} = \frac{m_i n_j}{M} b_2 , \qquad \text{[Eq. 4-9]}$$

where

$$b_2 \equiv \frac{N}{N + n_g} . \qquad \text{[Eq. 4-10]}$$

Noting the similarity of Equations 4-8 and 4-10, we introduced into SLAACM the simple model

$$\overline{E}_{ij} = \frac{m_i n_j}{M} b \qquad \text{[Eq. 4-11]}$$

and allow the user to choose the "battle management efficiency factor" b, where $0 < b \leq 1$.

Once seen, Equation 4-11 is so simple, and seemingly obvious, that it is fair to ask why we did not simply introduce it into SLAACM without analysis. The reason is that we wanted to understand what that simple choice implied about how the battle proceeded. The analysis given here supplies that understanding.

Chapter 5
How SLAACM Works: SLAACM Campaign Calculations

This chapter describes the calculation protocols used by SLAACM for campaign analysis.

DETERMINING AND PROPAGATING LOSS AND DESTRUCTION STATISTICS

The calculations described in Chapter 4, under “Determining Attacks and Defenses,” generate the sets of engagements that take place in each day’s fighting. There are then specified numbers of engagements between a specific Red attack package (with its specified aircraft types for the advanced escorts, close escorts, and bombers) and a flight of 4 defenders of a specific type. SLAACM uses the method described in Chapter 2, under “Composite Engagements,” to generate probabilities for all 16 outcome states of each kind of engagement that occurs.

Then, to evaluate Red losses, SLAACM calculates expected values of the losses for advanced escorts, close escorts, and bombers. These expected values are accumulated for each Red aircraft type, for engagements between smart Blues and the attack packages, and for engagements between non-smart Blues and the attack packages not engaged by smart defender aircraft.

The total expected losses for each Red type are rounded to integers and subtracted from the Red order of battle. SLAACM then generates the Red order of battle for the following day using the surviving Reds plus any reinforcements scheduled to arrive for that day. Reinforcements are a user input to SLAACM.

Expected values of surviving bombers are multiplied by bomber payloads and rounded to the nearest integer, to give tons-of-bombs results. Tons delivered by bombers with relative CEPs less than a user-specified value are identified as “smart” and accounted for separately from tons delivered by other bombers.

In treating Blue losses, SLAACM considers measures of dispersion as well as measures of central tendency. The probability distributions for outcome states of the engagements occurring each day are used to calculate both means and variances of the losses of each type of Blue aircraft. Mean losses, rounded to the nearest integer, are subtracted from Blue’s order of battle and—after increments from any scheduled reinforcements—are posted as Blue’s order of battle for the next day.

Variances in the losses of each type of Blue aircraft are accumulated day by day, and the total variances are reported to give a measure of dispersion in Blue losses.

This procedure neglects a potentially important source of dispersion: the effect of propagating losses other than the means of both Red and Blue losses. In the cases that we have considered, the Blue aircraft are always much stronger than the Red aircraft. Consequently, there is little dispersion in Red losses—nearly all Red aircraft that are engaged are lost, nearly all the time—so the significant dispersion in each day is the dispersion in Blue losses. But, again by virtue of Blue's much greater strength, the dispersions never lead to significant probabilities of Blue's losing substantial fractions of their forces, except near the ends of campaigns in which Blue is losing. The present dispersion calculations are useful for checking the validity of using only mean values. We intend to treat the impact of dispersion of day-to-day losses in future work.

SLAACM'S SEQUENCE OF CALCULATIONS

This section describes SLAACM's step-by-step calculations. It explains how SLAACM uses the methods described above to model air-to-air campaigns.

User Inputs

The SLAACM user inputs Red and Blue orders of battle for the first day of fighting and lists any scheduled reinforcements.[1] The following subsections summarize the inputs, focusing on their meanings and use. (Chapter 6 details the methods for entering data into SLAACM.)

INPUTS FOR THE BLUE FORCE

The user may identify certain Blue aircraft as smart. Smart Blue aircraft can determine the composition of Red attack packages before intercepting them, and can coordinate their defensive response among themselves to provide optimal defenses.

Blue aircraft based long distances from the battle space will have to be rationed and rotated to maintain continuous defensive coverage. The user may select a combat air patrol (CAP) factor for any Blue aircraft type, so that only a specified fraction of the inventory of that Blue type is available to meet a Red attack.

The ability of Blue aircraft that are not smart to find and reach the Red packages is determined by the battle management efficiency factor, b, discussed in Chapter 4. This factor has a user-defined value of $0 < b \leq 1$, where b corresponds to the fraction of the assigned Blue flights able to intercept Red packages.

[1] "Day" is used to signify a set of attack and defense sorties. There could be more than one set of sorties on a single calendar day, or there could be gaps of calendar days between attacks.

The user may identify certain Blue aircraft as local air defenders (LADs). SLAACM treats LADs differently from other Blue types when considering Blue dispatch options. Normally Blue values the life of it forces, and will not dispatch a Blue type if engaging a Red package gives a negative value of Blue's payoff function. Blue types designated as LADs, however, will be dispatched in all cases.

INPUTS FOR THE RED FORCE

The user inputs the bomb payload of each Red aircraft and specifies whether the aircraft is to be used as a fighter only, or as a bomber only, or as either a fighter or a bomber. The user also inputs the ratio of the circular error probable (CEP) with which bomber aircraft can deliver bombs, to the reference CEP. As discussed in Chapter 3, the default ratio is 1.0 for dumb bombs and less than 1.0 for smart bombs. If delivery of cruise missiles is of interest, the user inputs the number of cruise missiles that can potentially be carried by each aircraft.

Red's order of battle may be so large that airfield capacity and battle management capability may preclude using all forces. The user may specify a maximum number of Red aircraft that can be dispatched in a given attack.

Red optimizes its dispatch choices with the assumption that each package encounters a specified Blue fighter type. The user identifies this type.

INPUT FOR RELATIVE CAPABILITIES OF RED AND BLUE AIRCRAFT

Currently, SLAACM models engagements with the M vs. N probabilistic engagement model, and it assumes that outcome probabilities have their long-time limiting values as described in Chapter 2 under "Calculations for the Long-Time Limit." Thus, only one aircraft parameter—the ratio κ of the parameter of the Blue time-between-kill distribution to that parameter for the Red time-between-kill distribution—affects outcomes.

Values of κ parameters may not be readily available, and SLAACM includes a utility to calculate κ from the Blue to Red "loss ratio," that is the generally available output from simulations. (Some simulations provide exchange ratios, which are ratios of Blue relative losses to Red relative losses. Multiplying an exchange ratio by the ratio of the initial number of Blues to the initial number of Reds gives the loss ratio.) The user inputs the loss ratios and the source scenario, such as 4 Blues vs. 8 Reds with Blue breaking away after 2 losses. SLAACM then determines the value of κ that gives the observed loss ratio and uses it in subsequent calculations. SLAACM uses bisection to find κ values correct to four significant figures.

With all these inputs completed, the user enters the number of days to run the campaign, specifies the battle management efficiency parameter as described in Chapter 4 under "The Battle Management Efficiency Parameter" and instructs

SLAACM to use either heuristic optimization or exact optimization with LINGO. SLAACM then performs the sequence of calculations described below.

Steps in SLAACM's Calculations

SLAACM first calculates values of the Red payoff function, as described in Chapter 4 under "Red's Payoff Function," for all possible choices of attack escort, close escort, and bomber that can make up a Red assault package. There will often be hundreds of such choices. SLAACM stores the payoff values for use in later optimizations.

SLAACM then begins the day-by-day calculations of outcomes in the campaign. Using either the heuristic or the exact solution, as specified by the user, SLAACM determines the set of attack packages that maximizes the total Red payoff function, subject to constraints imposed by Red's available forces and to the maximum number of Red aircraft that can be dispatched in one day as specified by the user. These calculations are described in Chapter 4 under "Determining Attacks and Defenses."

SLAACM considers this set of attack packages to determine the optimal defense by the smart Blue aircraft. After calculating the Blue payoff, as described in Chapter 4 under "Blue Payoff Function," for each possible combination of smart Blue defense with a Red package type actually dispatched, SLAACM determines the Blue dispatch option that maximizes the total Blue payoff, constrained by the available Blue forces and the CAP values input by the user. SLAACM uses either the heuristic or the exact solution of the resulting IP problem, as directed by the user.

As described above under "Determining and Propagating Loss Statistics," SLAACM then evaluates the losses to all forces from engagements between the Red attack packages and the smart Blue defenders. Then SLAACM calculates the results of engagements among not-smart Blue defenders and the Red packages that were not engaged by smart Blue aircraft. Only those Red packages not engaged by smart defenders are available for engagements with the not-smart Blue aircraft.

SLAACM calculates the Blue payoff for each combination of not-smart Blue aircraft and a Red attack package type that was not intercepted by a flight of smart Blue defenders. Then, following the procedure described in Chapter 4 under "Random Engagements by Not-Smart Blue Aircraft," and considering the user's input of battle management efficiency factor, SLAACM determines the numbers of engagements between the several types of not-smart Blues and the Red package types available to them. As described above under "Determining and Propagating Loss Statistics," SLAACM updates the losses of all aircraft types, calculates the tons of "smart" and "not-smart" bombs delivered, and, in view of any scheduled reinforcements, updates Red and Blue orders of battle for the next day of the campaign.

After calculating outcomes for the number of days specified by the user, SLAACM produces daily records of Red and of Blue orders of battle, of Red losses, of Blue losses and their variances due to engagements, of tons of smart and of not-smart bombs delivered, and of cruise missiles delivered. As described in Chapter 6, SLAACM also produces graphs of Red and of Blue drawdowns, the quantities and types of bombs delivered, and information on “who shot whom.”

Chapter 6
Users Guide to SLAACM

SLAACM is housed in a Microsoft Excel workbook with extensive Visual Basic code. In addition, it has an option to conduct integer programming by linking to an external optimization program called LINGO.[1] Using Excel to house SLAACM has several advantages. Most analysts are familiar with the Excel environment and will be able to navigate it easily. Other than acquiring the SLAACM workbook, users do not need to install additional software unless they want to run integer programming optimizations in LINGO. Finally, users can easily copy and paste output values and charts into their preferred presentation formats for displaying results to others.

This chapter describes how to operate SLAACM. It begins with a description of the model inputs. It then describes how to run the model once all of the input parameters are set up. Next, the chapter describes the outputs reporting SLAACM's results. The last section addresses how the outputs are analyzed to determine the effects of the various engagement scenarios.

INPUTS

SLAACM's input parameters are defined in five worksheets:

- BlueSupply, which is used to specify the quantities of Blue aircraft available on each day of the campaign
- RedSupply, which is used to specify the quantities of Red aircraft available on each day of the campaign
- BlueOOB, which is used to define Blue's order of battle (SLAACM also has a RedOOB worksheet, but it is purely an output worksheet and does not receive any user inputs)
- ExRatios, which is used to specify engagement parameters such as loss ratios and breakpoints
- CM, which is used to specify the number of cruise missiles that a Red aircraft could carry.

We discuss each sheet below and describe the proper way to complete each one.

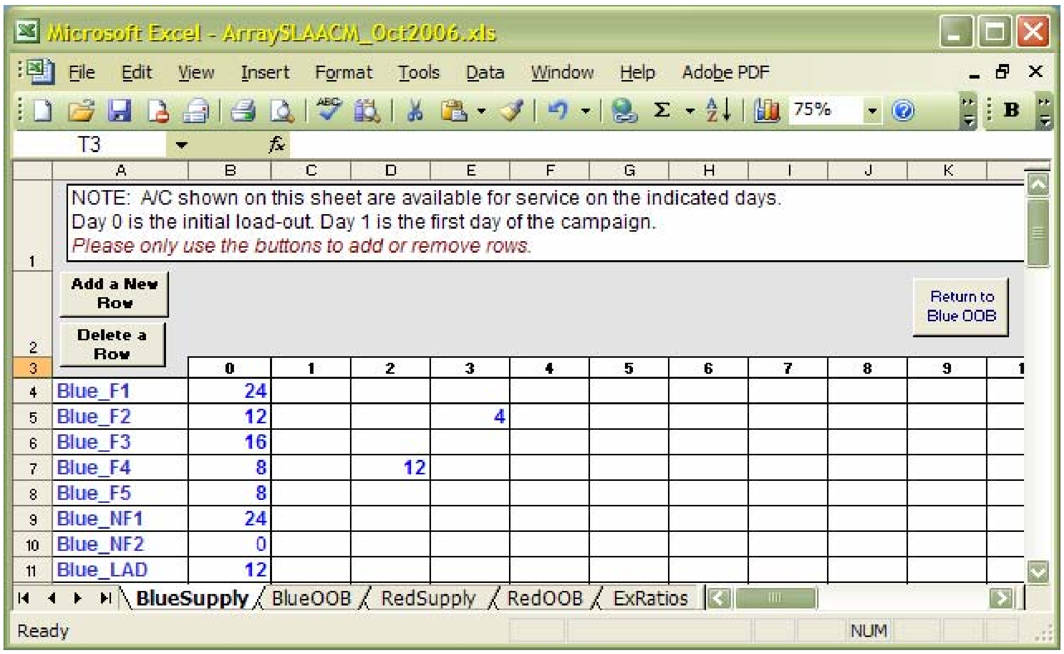
[1] For more information on LINGO, see http://www.lindo.com/products/lingo/.

BlueSupply and RedSupply Worksheets

For each force, Blue and Red, the user will need to set up the initial quantities of aircraft available and the reinforcements available on each day of the campaign. The user will need to type those quantities into the appropriate worksheets, called BlueSupply and RedSupply.

Figure 6-1 is an example of the BlueSupply worksheet. As shown in the figure, the worksheet has a row for each Blue aircraft type and a column for each day of the campaign. Initial aircraft quantities are entered into the column labeled 0. Reinforcements may be provided for subsequent days of the campaign. In the figure, we see that reinforcements are entered for Day 2 and Day 3.[2]

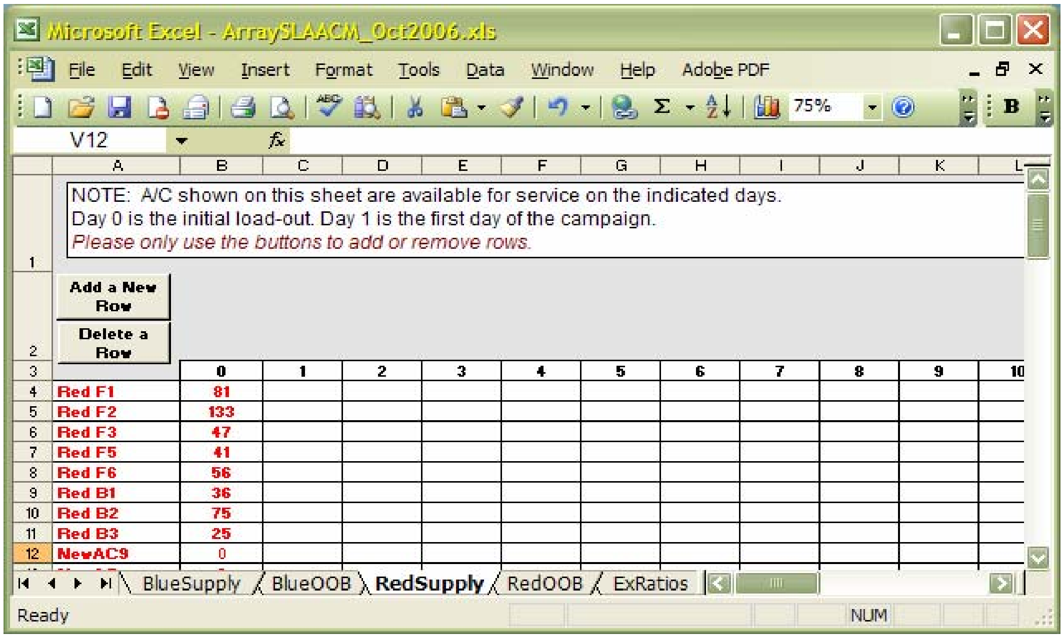
Figure 6-1. BlueSupply Worksheet



The RedSupply worksheet is designed in the same way as the BlueSupply sheet. An example is shown by Figure 6-2. In the figure, we see that only initial aircraft quantities are provided and no reinforcements are specified.

[2] Note that a day is used in the model as a convenient designator to represent an individual attack/defense sortie. In an actual campaign there may be more than one sortie on a particular calendar day, or, conversely, sorties may be distributed among several calendar days.

Figure 6-2. RedSupply Worksheet



The BlueSupply and RedSupply worksheets can accommodate up to 20 aircraft types each. If new rows are needed to accommodate additional aircraft types, they can be added using the “Add a New Row” button.

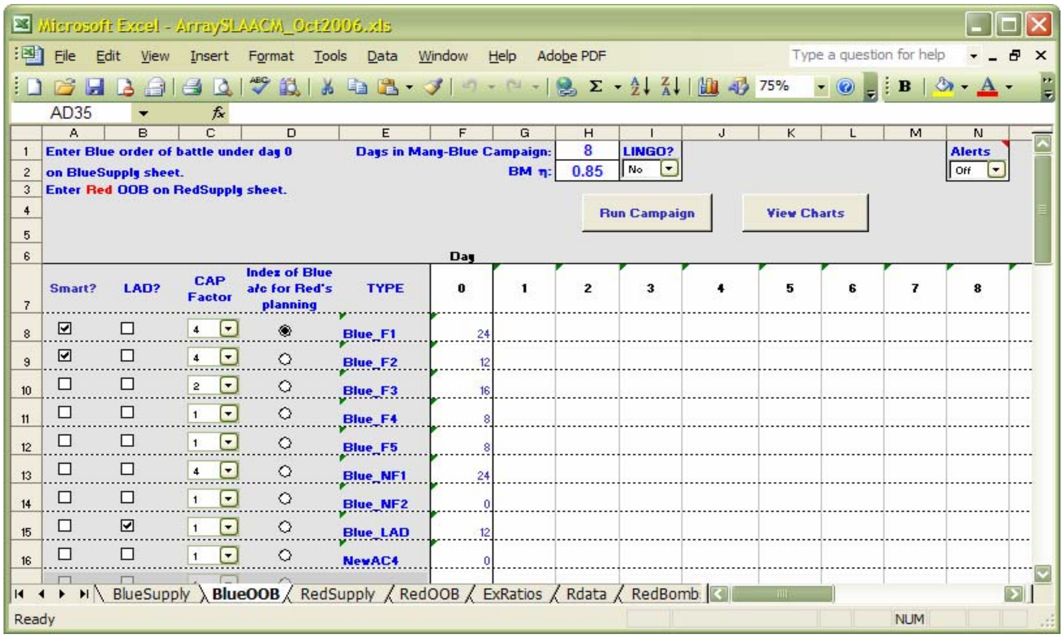
Caution!
Any time a row is added or deleted it is imperative that the ***buttons*** *on the BlueSupply and RedSupply worksheets are used to add or delete the rows.* ***Do not*** *simply insert or delete rows and columns.*

Several of SLAACM’s worksheets reference the aircraft types and the “Add a New Row” and “Delete a Row” buttons must be used to keep the worksheets synchronized, Using typical worksheet approaches to inserting and deleting rows will likely result in model errors or misinterpretation of results.

BlueOOB Worksheet

The worksheet named BlueOOB (which stands for Blue order of battle) is the main worksheet for input parameters for the Blue aircraft. It is also the worksheet used to launch campaign analyses. BlueOOB also serves as an output worksheet, as does the RedOOB worksheet; the output portion of BlueOOB will be discussed in the output section of this chapter. Figure 6-3 shows the BlueOOB worksheet.

Figure 6-3. BlueOOB Worksheet

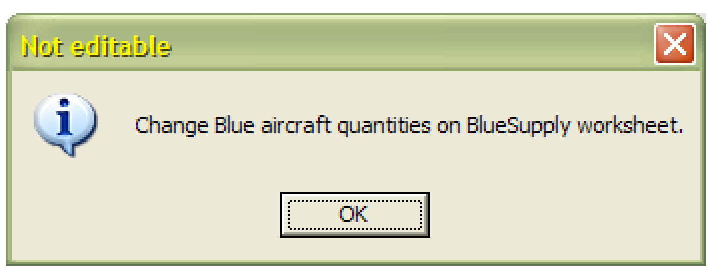


Caution!

Rows or columns should never be deleted from this worksheet!

The aircraft listed in the BlueOOB worksheet are linked to those that were shown in the BlueSupply worksheet of Figure 6-1. The Day 0 quantities are also linked to those in the BlueSupply worksheet. The user who attempts to click the aircraft names or Day 0 quantities on the BlueOOB sheet will receive the warning shown in Figure 6-4.

Figure 6-4. Warning: Do Not Alter Aircraft of BlueOOB Worksheet



The first four columns on the BlueOOB worksheet define aircraft-specific input parameters for Blue:

- *"Smart?" column.* The user may use this column to identify specific Blue aircraft as smart. Smart Blue aircraft can determine the composition of Red attack packages before intercepting them and can coordinate their defensive response among themselves to provide optimal defenses. Coordination and optimization are discussed in Chapter 5. Multiple Blue aircraft types may be designated as smart. In Figure 6-3, the first two aircraft are selected to be smart.

- *"LAD?" column.* The user may use this column to identify certain Blue aircraft as local air defenders. SLAACM treats LADs differently from other Blue types when considering Blue dispatch options. In general, a non-LAD Blue has a penalty for its own losses that is significantly greater than a LAD Blue, which presumably is less loss-averse due to defending its own territory. Some non-LAD Blues may have a negative payoff function for engaging certain Red packages dispatched and Blue will not dispatch that type. LADs, however, are dispatched in all cases. Multiple Blue aircraft types may be designated as LADs. In Figure 6-3, the eighth aircraft is designated as such. Note that SLAACM also has a separate aircraft type labeled Blue_LAD, which is intended to represent an indigenous "Green" defender. In earlier versions of SLAACM, this Blue_LAD was the only aircraft in the model that placed no value on its own survival. The current version allows all aircraft to be optionally "fearless." This facilitates the analysis of supplying different types of aircraft to the indigenous forces. (We plan to address any confusion in the LAD and Blue_LAD terminology in future versions of SLAACM.)

- *"CAP Factor" column.* This factor addresses situations in which Blue fighters are stationed at remote airfields and must divide their forces to maintain continuous combat air patrol coverage over the battle space. The user may select a CAP factor for any Blue aircraft type, so that only a specified fraction of the inventory of that Blue type is available. CAP factor values are chosen from a drop-down box for each aircraft type. Values range from 1 to 10. A CAP factor of 1 indicates that no CAP limitation is placed on that aircraft type. In Figure 6-3, several aircraft have CAP factors greater than 1. For example, the first aircraft has been assigned a CAP factor of 4. That aircraft, Blue_F1, also begins with an initial quantity of 24 aircraft. Therefore, Blue_F1 has only 24/4 = 6 available aircraft on the first day, which means that only one 4-ship package of Blue_F1s can be deployed. In addition, if Blue_F1 losses on subsequent campaign days were to exceed 8 aircraft, Blue could no longer dispatch Blue_F1s, since 15/4 < 4. Thus, Blue_F1 would be unable to maintain a 4-ship package at all times. This assumption is fairly conservative: if there are not enough aircraft to fulfill the combat air patrol at all times, those flights are not dispatched.

- "Index of Blue a/c for Red's planning" column. Only one aircraft type can be selected for this parameter. In Figure 6-3, the first aircraft type is designated as Red's "planning aircraft." This aircraft type is used by Red to compute its expected payoff function and optimize its dispatch choices. Experience has shown that Red optimization is relatively insensitive to the selection of the Blue type, so normal practice is to select the most numerous Blue type for Red planning.

In addition to the aircraft specific parameters, several other input parameters must be selected on the BlueOOB worksheet.

At the top of the worksheet, the number of days in the campaign is specified. In Figure 6-3, the number of campaign days is 8. The longer the campaign, the longer it will take SLAACM to run, since each campaign day requires the full breadth of SLAACM computations. Initially, the user may need to start out with a longer campaign and then decrease its length as a steady state in aircraft losses is reached. Eight is a good initial number to use since some campaigns will complete earlier and some later, depending on the mixes and capabilities of Blues and Reds specified.

Beneath the number of campaign days is a parameter labeled "BM η." This parameter is the battle management efficiency factor. BM η is a value greater than 0 and less than or equal to 1. Simply stated, a BM η less than 1 indicates the fraction of time that Blue defender will be able to find a Red attack package. In Figure 6-3, the efficiency factor is shown as 0.85, which means that 15 percent of the not-smart Blue defenders will chase "ghost" attackers. No losses will result, but in those cases, Blue will not be confronting an actual attacker, so its efficiency is reduced. Since smart Blues know Red's attack package compositions in advance, we assume they have perfect management.

Next to the campaign days and the efficiency factor is a drop-down box labeled "LINGO?" If "No" is selected, as shown in Figure 6-3, SLAACM will use its internal heuristic optimization to calculate engagement results. If "Yes" is selected, LINGO will be used to perform integer optimization (a licensed copy of LINGO software is required to be installed on the PC for this feature to work). Both the heuristic and the exact integer programming solution are described in detail in Chapter 5. If LINGO is used, there are two LINGO input worksheets: "RedOpt" and "BlueOpt." However, those sheets are already complete and should not be altered by the user.

The final input selection on BlueOOB is a drop-down box labeled "Alerts." The default setting for this feature is "Off." However, if "On" is selected, message boxes will become visible as the analysis runs.

ExRatios Worksheet

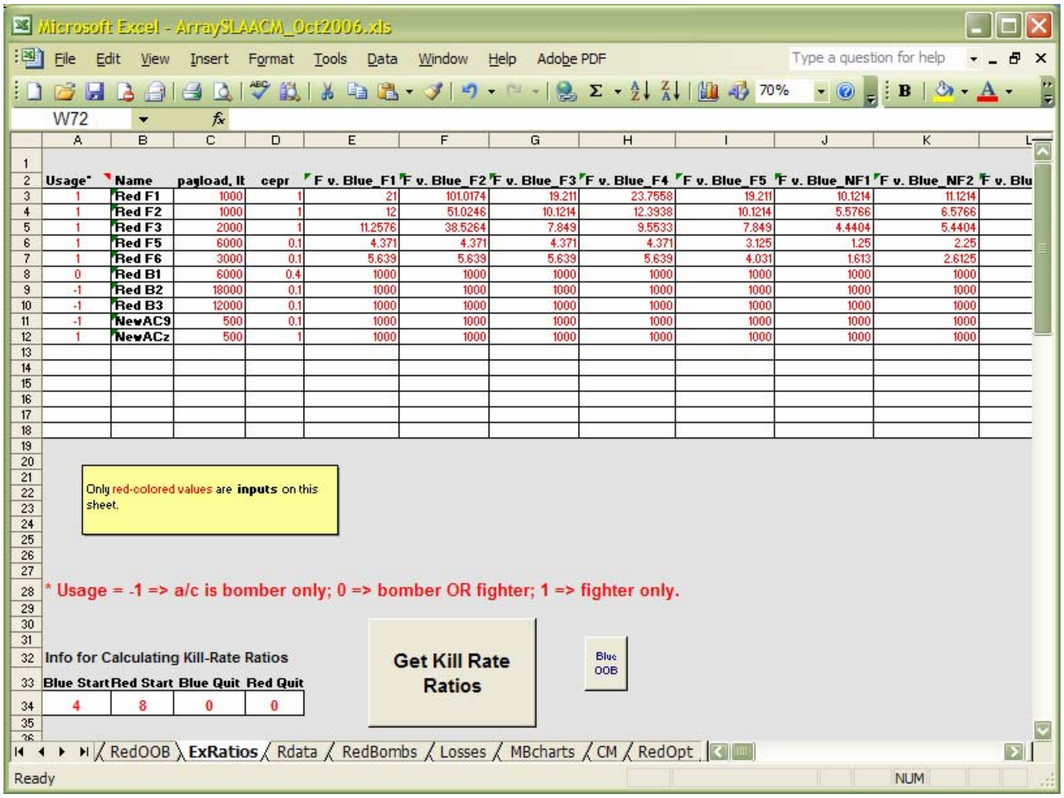
The ExRatios worksheet contains a row for each Red aircraft that has been designated on the RedSupply worksheet. The worksheet has several parameter columns, discussed below, and a column for each Blue aircraft labeled “F v. Blue_x” where “F” indicates that Red is a fighter and “x” corresponds to the label provided on BlueSupply. Columns are repeated for each Blue aircraft, except they are labeled “B v. Blue_x” where “B” indicates that Red is a bomber and “x” is as just described. The default setting in SLAACM assumes that loss ratios are for engagements between 4 Blues and 8 Reds, with the Reds and Blues fighting to annihilation. However, those assumptions can be changed on this sheet under “Red Start,” “Blue Start,” “Red Quit,” and “Blue Quit.” Red Start and Blue Start correspond to the engagement type, e.g., 8 and 4, respectively. Red Quit and Blue Quit denote the breaking points, e.g., 0 and 0 means that both sides will fight to annihilation. The user will want to specify values that correspond to the conditions under which the loss ratios were obtained.

To keep the rows and columns synchronized with the BlueSupply and RedSupply worksheets, it is imperative that no rows or columns on ExRatios be inserted or deleted.

Caution!
As stated earlier, use the “Add a New Row” and “Delete a Row” buttons on the BlueSupply and RedSupply worksheets to add or delete rows.

Figure 6-5 is a snapshot of the ExRatios worksheet.

Figure 6-5. ExRatios Worksheet



The first column on the ExRatios worksheet is labeled “Usage.” For each Red aircraft type, a value of 1, −1, or 0 needs to be entered: 1 indicates that the Red aircraft is a fighter, −1 indicates that the Red aircraft is a bomber, and 0 indicates that the Red aircraft can be used as both a fighter and a bomber. The second column indicates the name of the Red aircraft type; it is linked to the RedSupply worksheet. The third column, “Payload, lb,” is used to enter the number of pounds of bombs that the associated Red aircraft can carry.

The fourth column is labeled “cepr,” which stands for the circular error probable ratio (CEP/CEP_{ref}). The cepr is a value greater than 0 and less than or equal to 1, where 1 indicates that the bomber cannot deliver smart bombs (guided munitions). Values of cepr less than 1.0 indicate increasing accuracy of precision guidance. Bomber payoff values used by both Red and Blue are calculated from the payload tonnage and the weapon accuracy as defined by the cepr value. The tonnage defined as smart bombs in the SLAACM plotting routine is based on a threshold value contained on the worksheet named “RedBombs.”

The remaining “F v. Blue_x” and “B v. Blue_x” columns are filled with loss ratio values. Those values are generally determined from experience or simulation, and are beyond the scope of a user’s guide. It is expected that users will have access to relevant loss ratio data. Once the loss ratios are completed, the large button at the bottom of the screen labeled “Get Kill Rate Ratios” needs to be clicked. That button will run a macro that uses the loss ratios to compute kill rate ratios. The kill

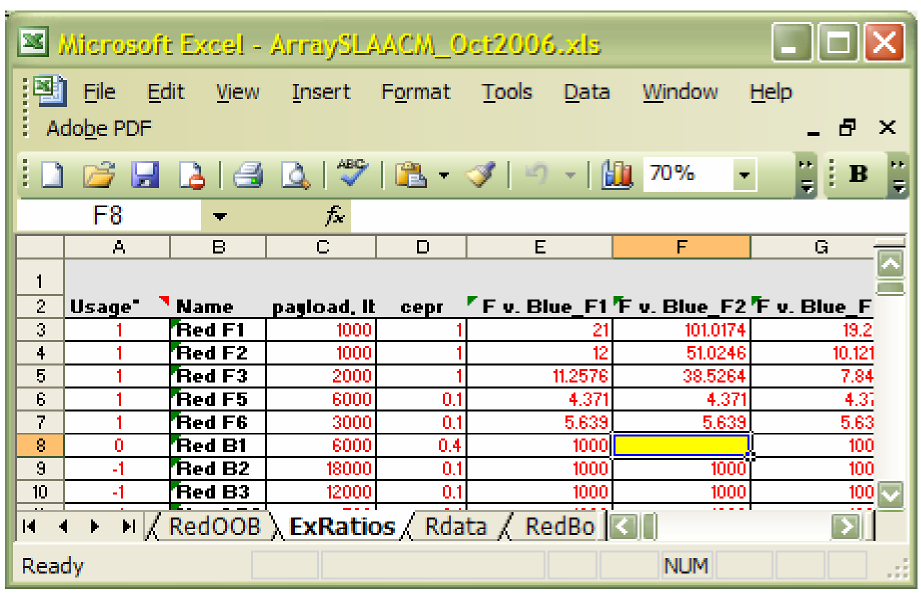
rate ratios are automatically entered onto the worksheet labeled "Rdata." As an added safeguard, every time SLAACM is run, the kill rate ratio macro is run to ensure that the most recent values are entered onto Rdata.

SLAACM will not run if the ExRatios worksheet contains any blank values. If a loss ratio is left blank on ExRatios, the error message shown in Figure 6-6 will appear.

Figure 6-6. Loss Ratio Error

After clicking "OK," the user will be taken to the ExRatios worksheet and any blank values will be highlighted as shown in Figure 6-7.

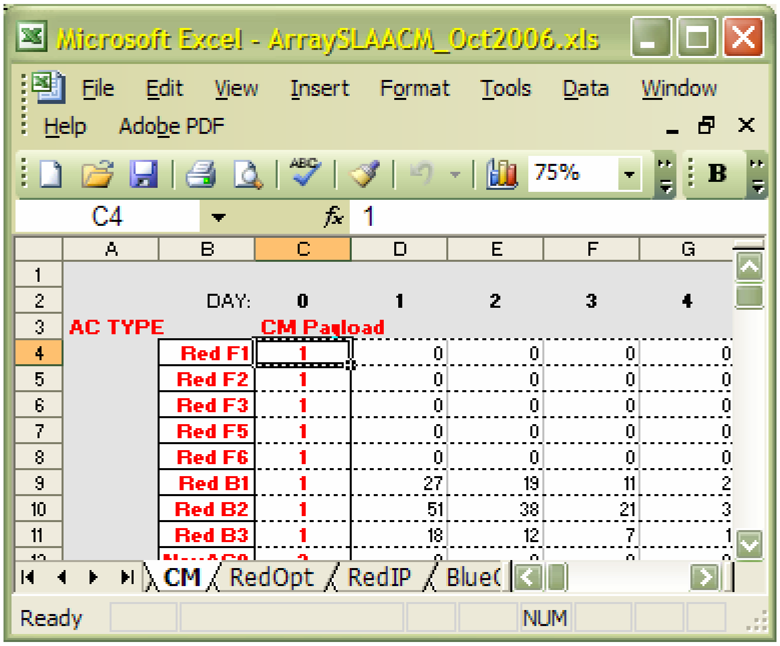
Figure 6-7. Blank Loss Ratio Highlighted



CM Worksheet

The final user inputs are on the worksheet named "CM," which stands for cruise missiles. On this worksheet, the user can specify the number of cruise missiles a Red aircraft could carry. Figure 6-8 shows that column C is used to input the cruise missile capacity of each aircraft.

Figure 6-8. CM Worksheet



No tradeoff between cruise missiles and bomb payload is required because the cruise missile computation is a secondary analysis that is meant to provide "what-if" information to the user. In this computation, the number of Red aircraft or each type that get past Blue defenses each day is multiplied by the number of cruise missiles specified on the CM worksheet. This analysis merely gives an idea of the potential number of cruise missiles that Red could launch if its aircraft were armed to do so.

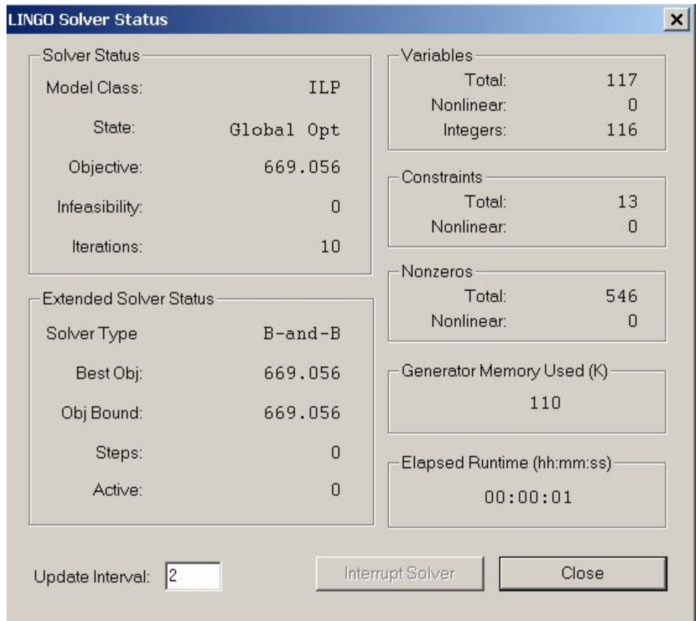
RUNNING SLAACM

Once all of the inputs have been entered, the user needs to return to the BlueOOB worksheet. To launch the analysis, the user clicks the "Run Campaign" button.

If the heuristic is being used, no other action is required on the part of the user.

If LINGO is being used, then a series of message boxes will appear to let the user know what day is being considered and which optimization is taking place. The user simply needs to click "OK" to proceed. The optimization in LINGO usually takes only a couple of seconds and the screen often clears automatically. If it does not clear by itself, the user may need to click "Close," provided LINGO has solved for the globally optimal solution (noted by "Global Opt" in the "State"). This case is shown in Figure 6-9.

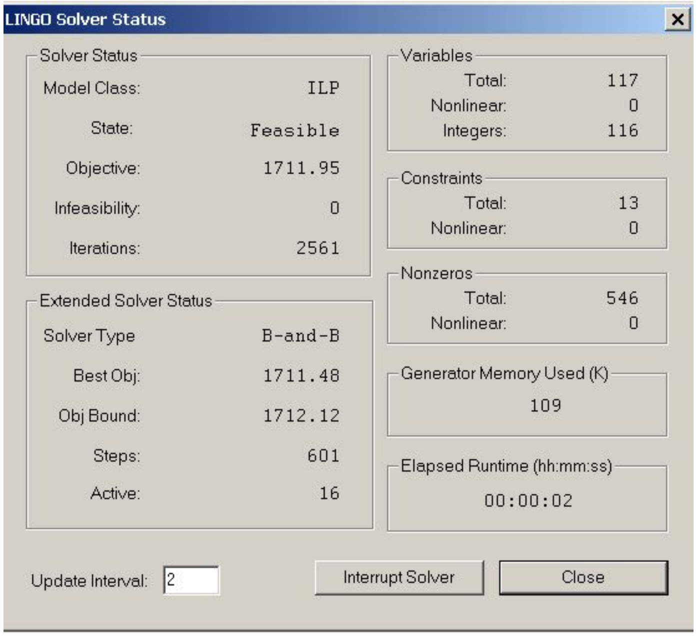
Figure 6-9. LINGO Solution for Red Optimization



If the "State" shows "Feasible" as its status, then the user should wait until the global optimal is reached. If the solver is running for a long time, and the difference between the "Best Obj" and "Obj Bound" is small (e.g., less than 0.5%), then the user may click "Interrupt Solver" and the best feasible solution at that point will be entered into SLAACM. This event occurs rarely in SLAACM, since LINGO is usually able to solve these integer programming problems very quickly.

Figure 6-10 shows LINGO before it has reached optimality; it has reached a feasible—but not necessarily optimal—solution. In this example, the difference between the best objective and the objective bound is less than 1/1,712 = 0.05%, so one could certainly interrupt the solver at this point.

Figure 6-10. Feasible LINGO Solution



OUTPUTS

Once the analysis has been completed, the user has several output results to review. SLAACM produces daily records of Red and Blue orders of battle, of Red and Blue losses, of their variances due to engagements, of tons of smart and not-smart bombs delivered, and of cruise missiles delivered.

Six worksheets report SLAACM's output results:

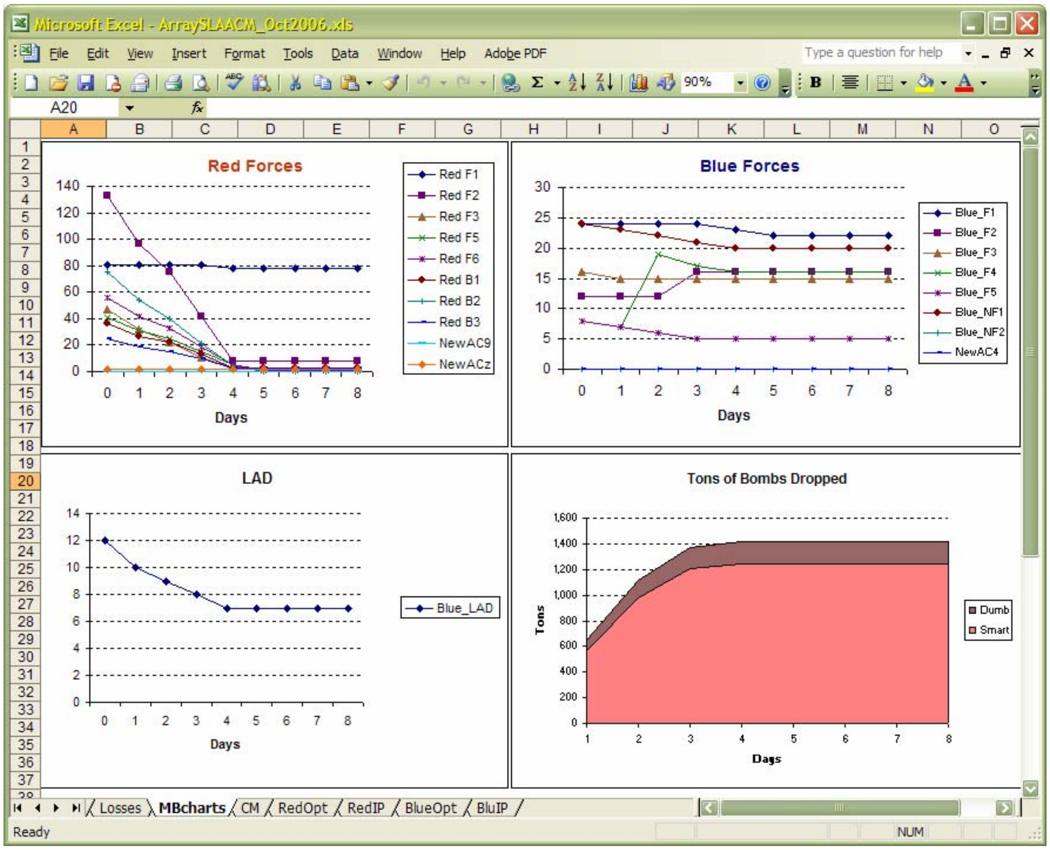
- BlueOOB
- RedOOB
- RedBombs
- Losses
- MBCharts
- CM.

We discuss the outputs—charts and worksheets tables—below. By reviewing the charts and tables, the user can gain an understanding of specific campaign outcomes.

Charts

The main results are presented graphically. From the BlueOOB worksheet, the user can click the “View Charts” button, or simply proceed to the worksheet tab labeled “MB Charts.” Figure 6-11 shows the quad chart “dashboard” of the main SLAACM results.

Figure 6-11. Output Charts



The “Red Forces” chart in the upper left corner of Figure 6-11 shows the quantity of Red aircraft on each day of the campaign. In the example, most Red aircraft are annihilated by Day 4. The aircraft that have some quantity remaining, but which reach steady-state quantities, are most likely not being sent. For example, Red F1 is a weak fighter, so Red is choosing to escort its bombers with more capable aircraft. A quick check of the “Tons of Bombs Dropped” chart in the lower right corner shows that it looks like the bomb levels have stopped growing by Day 4, indicating that no Red bombers are getting through Blue defenses beyond that day. That leads us to believe that the campaign is over by Day 4.

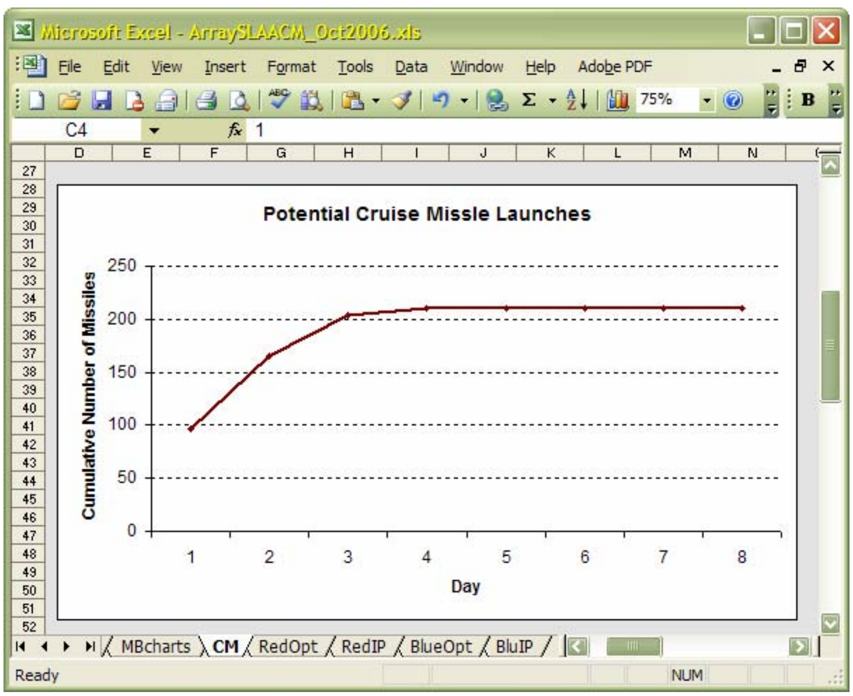
The “Blue Forces” chart in the upper right corner of Figure 6-11 shows the daily quantities of Blue aircraft. We see that Blue sustains some losses, although not nearly as many as Red; therefore, we can assume that Blue’s forces are superior to Red’s. In addition, in the first few days, we see increases in some quantities of

Blue aircraft. That is due to Blue receiving reinforcements. (See the BlueSupply worksheet in Figure 6-1.) Blue_LADs are broken out separately from the Blue Forces and are shown in the chart in the lower left corner of Figure 6-11.

The Red Forces, Blue Forces, and LAD charts show the quantity of each aircraft type that is *available* on each day of the campaign. The charts do not indicate the number of aircraft *sent* into battle.

Figure 6-12 shows a graph of the potential number of cruise missile launches. As discussed in the previous section, this result assumes that the Red bombers that got through Blue defenses were armed with cruise missiles in lieu of bombs and were able to launch their cruise missiles.

Figure 6-12. Cruise Missile Launches



Worksheet Tables

Because SLAACM is housed in an Excel workbook, the user can browse the data used to build the charts. All of the charts in Figure 6-11 are built from data in worksheet tables:

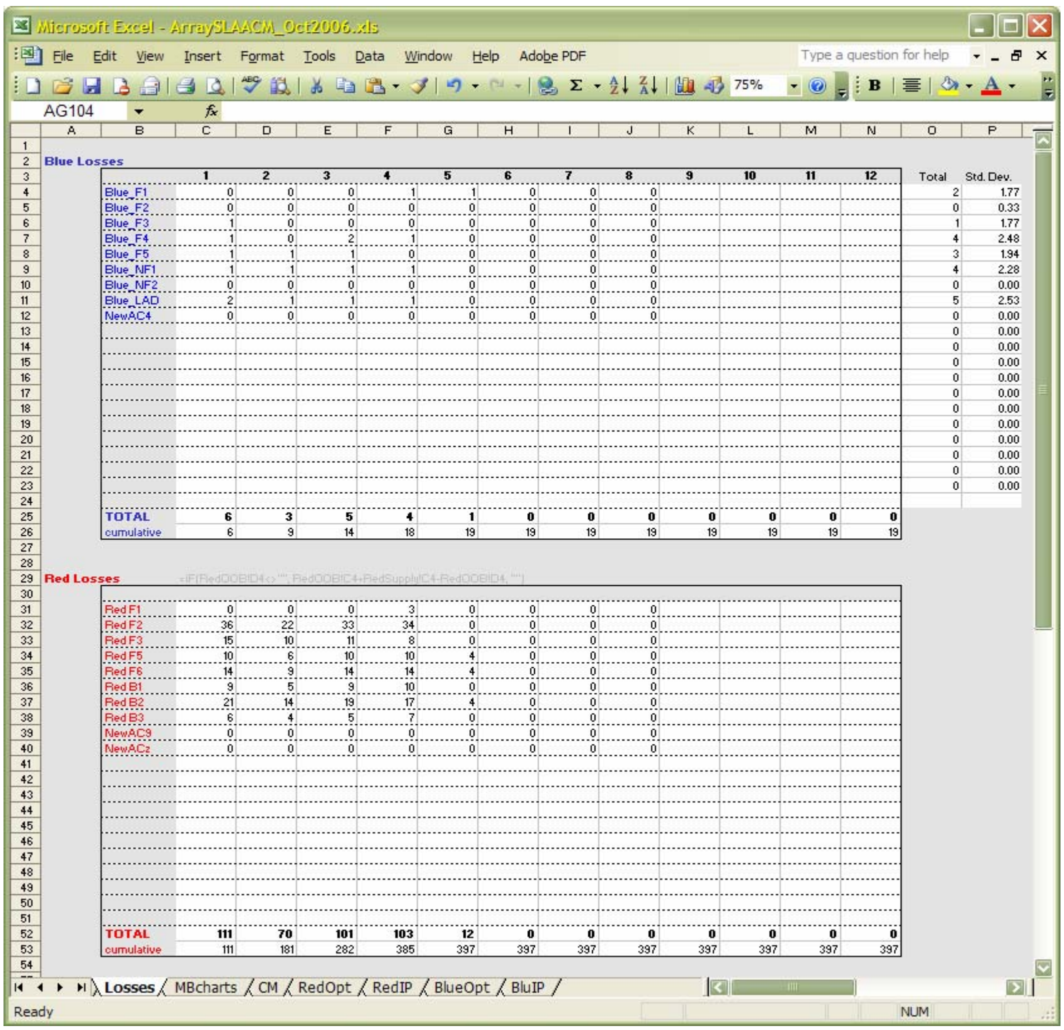
- The Red Forces chart uses the data on the “RedOOB” worksheet.
- The Blue Forces and LAD charts use data from the “BlueOOB” worksheet. (As we discussed earlier, the BlueOOB worksheet is both an input

and an output. After running SLAACM, available aircraft quantities are loaded into BlueOOB worksheet for each day of the campaign.)

- The Tons of Bombs chart uses data from the “RedBombs” worksheet.
- The Cruise Missile chart is built from data on the “CM” worksheet, where the chart resides.

It can be helpful to look at aircraft losses to get a better idea of how the campaign progressed. SLAACM has a worksheet named “Losses” that shows the loss data. Figure 6-13 shows the Losses worksheet. Blue and Red losses are shown by aircraft type. In addition, the worksheet shows the standard deviation of the losses for Blue aircraft.

Figure 6-13. Losses



Blue Losses

	1	2	3	4	5	6	7	8	9	10	11	12	Total	Std. Dev.
Blue_F1	0	0	0	1	1	0	0	0					2	1.77
Blue_F2	0	0	0	0	0	0	0	0					0	0.33
Blue_F3	1	0	0	0	0	0	0	0					1	1.77
Blue_F4	1	0	2	1	0	0	0	0					4	2.48
Blue_F5	1	1	1	0	0	0	0	0					3	1.94
Blue_NF1	1	1	1	1	0	0	0	0					4	2.28
Blue_NF2	0	0	0	0	0	0	0	0					0	0.00
Blue_LAD	2	1	1	1	0	0	0	0					5	2.53
NewAC4	0	0	0	0	0	0	0	0					0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
													0	0.00
TOTAL	**6**	**3**	**5**	**4**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
cumulative	6	9	14	18	19	19	19	19	19	19	19	19		

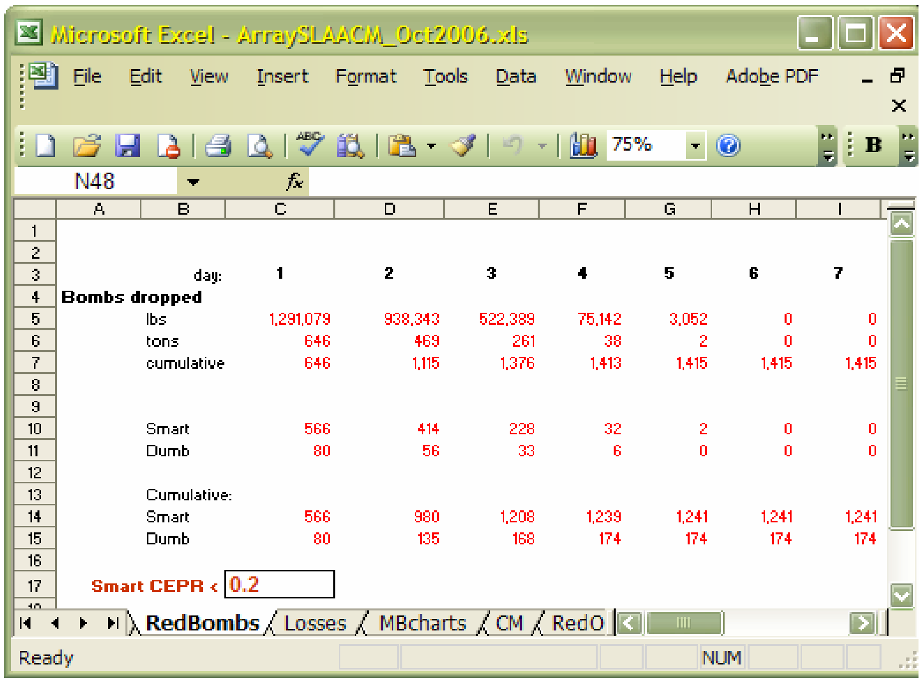
Red Losses

Red F1	0	0	0	3	0	0	0	0				
Red F2	36	22	33	34	0	0	0	0				
Red F3	15	10	11	8	0	0	0	0				
Red F5	10	6	10	10	4	0	0	0				
Red F6	14	9	14	14	4	0	0	0				
Red B1	9	5	9	10	0	0	0	0				
Red B2	21	14	19	17	4	0	0	0				
Red B3	6	4	5	7	0	0	0	0				
NewAC9	0	0	0	0	0	0	0	0				
NewACz	0	0	0	0	0	0	0	0				
TOTAL	**111**	**70**	**101**	**103**	**12**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
cumulative	111	181	282	385	397	397	397	397	397	397	397	397

Figure 6-14 shows the pounds and tons of bombs dropped by Red aircraft each day of the campaign. In addition, the bomb types are broken out by smart and not-smart types. These are the data used to build the “Tons of Bombs Dropped” chart

in the lower right corner of Figure 6-11. Upon inspection of the data in Figure 6-14, we see that the campaign actually ends on Day 5, not Day 4 as it had appeared on the chart; one can see that 2 tons of bombs were dropped on Day 5.

Figure 6-14. Red Bombs



	day:	1	2	3	4	5	6	7
Bombs dropped								
	lbs	1,291,079	938,343	522,389	75,142	3,052	0	0
	tons	646	469	261	38	2	0	0
	cumulative	646	1,115	1,376	1,413	1,415	1,415	1,415
	Smart	566	414	228	32	2	0	0
	Dumb	80	56	33	6	0	0	0
	Cumulative:							
	Smart	566	980	1,208	1,239	1,241	1,241	1,241
	Dumb	80	135	168	174	174	174	174
Smart CEPR <		0.2						

User Note

The output worksheets in SLAACM are dynamic and are overwritten during each model run; users who want to save run data should copy and paste the values into an external file.

Analyses of Sample Runs

In this section, we conduct a comparative analysis of the heuristic and LINGO using two SLAACM runs to illustrate some of the subtle differences that occur under reasonably similar scenarios.

First, we run SLAACM using the inputs shown in Figures 6-1 through 6-5 using the heuristic; we then repeat the run using LINGO. As shown in Table 6-1, the results are identical; the heuristic performs the same as the LINGO optimization. This case is not uncommon, and it gives us confidence that the heuristic can be a

good approximation for the optimal solution (the heuristic and the exact integer programming formulations are covered in detail in Chapter 4).

Table 6-1. Results of Initial SLAACM Run

Item	Heuristic	LINGO
Tons dropped		
Smart bombs	1,241	1,241
Dumb bombs	174	174
Aircraft losses		
Blue	19	19
Red	397	397

To illustrate a situation in which the heuristic and LINGO solutions differ, consider the same scenario, except that all Blue aircraft are smart. This change requires checking the boxes on the BlueOOB worksheet (Figure 6-3). The results are shown in Table 6-2. For both model runs, the tonnage of smart bombs dropped is reduced by almost 50 percent, demonstrating the impact of smart Blues preferentially engaging the highest value Red packages. Table 6-2 also shows that the heuristic results and LINGO's exact IP results show significant differences.

Table 6-2. Results of Setting All Blues to Be Smart

Item	Heuristic	LINGO
Tons dropped		
Smart bombs	640	602
Dumb bombs	289	289
Aircraft losses		
Blue	19	17
Red	402	404

LINGO's optimal solution resulted in 38 fewer tons of smart bombs being dropped, 2 fewer Blue aircraft losses, and 2 additional Red aircraft losses. In short, the IP solution using LINGO resulted in Blue reducing the amount of damage while sustaining fewer losses. This example illustrates what we have seen in other results, as well as what one might expect. The heuristic and the exact solution to the IP problem in LINGO generally produce qualitatively the same results. However, if one is looking for very precise results, the heuristic may fall short of true optimality by several percentage points.

SUMMARY

SLAACM is a fast, flexible, robust model that contains the key input parameters necessary to define the characteristics of a realistic air-to-air campaign. Model results in both tabular and graphical format clearly display the impacts of parameter choices and provide insight into campaign scenarios. The Excel workbook implementation allows results to be easily copied and pasted to other applications for reporting and presentation.

The above examples show differences in results using the heuristic optimization and solving the exact IP formulation with LINGO. In the many practical cases we have analyzed, it has been rare for the heuristic results to differ from the LINGO results by more than 10 percent, but careful analyses may require the use of exact solution methods, especially for actual campaign planning.

Appendix A
Direct Computation of Long-Time Limiting Probabilities of Boundary States

The structure of the evolution equations for probability distributions of the engagements we consider makes it possible to find the long-term limits of the absorbing boundary states directly, in finitely many steps, without formally solving those differential equations. In some interesting cases, the finite steps may be carried out by straightforward iteration. This appendix explains these facts and gives some examples.

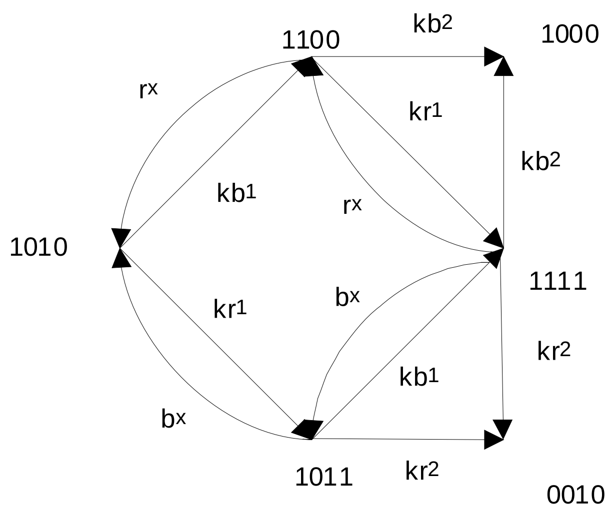
The evolution equations for the state probabilities in our engagement models decompose naturally into a set of equations for the transient states and a set for the absorbing boundary states. Equations 2-5 and 2-6 are examples of this for the basic M vs. N probabilistic engagement model. The advanced engagement models of Appendix D also have this property, except for the low-observable model, which is treated completely by the method given in Appendix D.

If we make the probabilities of the transient states the components of a vector x, and the components of the absorbing boundary states into the components of a vector y, the evolution of the state probabilities can be described by

$$\begin{aligned} \dot{x} &= Ax \text{ , } x(0) = x_0 \\ \dot{y} &= Bx \end{aligned} \quad . \qquad \text{[Eq. A-1]}$$

To illustrate these ideas for an example that, unlike the one of Chapter 2, has recurrent states, let us consider treating a 1 vs. 1 engagement with a two-phase kill model with break-lock. We use the state description (m, i; n, j) where m denotes the number of Blue aircraft, i the number of Blues tracking opponents, n the number of Red aircraft, and j the number of Reds tracking opponents. Figure A-1 is a diagram of the states in this engagement and their transitions.

Figure A-1. Diagram of 1 vs. 1 Engagement



The evolution equations are

$$
\begin{aligned}
\dot{P}_{1010} &= -(kb_1 + kr_1)P_{1010} + b_x P_{1011} + r_x P_{1110} \\
\dot{P}_{1110} &= -(kb_2 + kr_1 + r_x)P_{1110} + kb_1 P_{1010} + b_x P_{1111} \\
\dot{P}_{1011} &= -(kr_2 + kb_1 + b_x)P_{1011} + kr_1 P_{1010} + r_x P_{1111} \\
\dot{P}_{1111} &= -(kb_2 + kr_2 + r_x + b_x)P_{1111} + kr_1 P_{1110} + kb_1 P_{1011} \\
\dot{P}_{1000} &= kb_2(P_{1110} + P_{1111}) \\
\dot{P}_{0010} &= kr_2(P_{1011} + P_{1111})
\end{aligned}
. \qquad \text{[Eq. A-2]}
$$

The first four of these describe the transient states, and the last two describe the absorbing boundary states. Defining x_1 as P_{1010}, x_2 as P_{1110}, x_3 as P_{1011}, and x_4 as P_{1111}, and defining y_1 as P_{1000}, y_2 as P_{0010}, we see that, in this case, the matrices A and B of Equation A-1 are

$$
A = \begin{pmatrix} -(kb_1 + kr_1) & r_x & b_x & 0 \\ kb_1 & -(kb_2 + kr_1 + r_x) & 0 & b_x \\ kr_1 & 0 & -(kr_2 + kb_1 + b_x) & r_x \\ 0 & kr_1 & kb_1 & -(kr_2 + kb_2 + r_x + b_x) \end{pmatrix} \qquad \text{[Eq. A-3]}
$$

and

$$
B = \begin{pmatrix} 0 & kb_2 & 0 & kb_2 \\ 0 & 0 & kr_2 & kr_2 \end{pmatrix}. \qquad \text{[Eq. A-4]}
$$

Now, the solutions of systems of ordinary differential equations of the form $\dot{x} = Ax$ for constant coefficient matrices A can always be expressed as finite linear combinations of generalized exponential functions of the time, that is, functions of the form $t^j e^{\lambda t}$, where the j are positive integers and the λ are the eigenvalues of the matrix A.[1] If the transients are, in fact, transient, then the real parts of the λ are all strictly less than 0. It follows that the $\hat{x}_i$, defined by

$$\hat{x}_i \equiv \int_0^\infty x_i(t)dt\,, \qquad \text{[Eq. A-5]}$$

exist. Then, on integrating the differential equations of Equation A-1 from 0 to ∞, using the initial condition on x and y, and remembering that the $x_i(t)$ tend to zero as $t \to \infty$, we find

$$\begin{aligned} -x_0 &= A\hat{x} \\ y_{\lim} &= B\hat{x} \end{aligned}\,, \qquad \text{[Eq. A-6]}$$

where $y_{\lim}$ denotes the limit of y as $t \to \infty$. It follows that

$$y_{\lim} = -BA^{-1}x_0\,. \qquad \text{[Eq. A-7]}$$

Equation A-7 expresses the long-time limiting values of the absorbing boundary state probabilities as the result of finite operations, that is, matrix inversion and multiplication.

Let us continue to illustrate these concepts with the 1 vs. 1 example considered above. For a numerical example, we take $kr_1 = 1$, $kr_2 = 2$, $r_x = 1$, $kb_1 = 3$, $kb_2 = 4$, and $b_x = 2$. Then the matrices A and B take the values

$$A = \begin{pmatrix} -4 & 1 & 2 & 0 \\ 3 & -6 & 0 & 2 \\ 1 & 0 & -7 & 1 \\ 0 & 1 & 3 & -9 \end{pmatrix} \qquad \text{[Eq. A-8]}$$

and

$$B = \begin{pmatrix} 0 & 4 & 0 & 4 \\ 0 & 0 & 2 & 2 \end{pmatrix}. \qquad \text{[Eq. A-9]}$$

It is as well to check that the eigenvalues of A are distinct and negative. Direct calculation shows that this is in fact the case: the eigenvalues are approximately

[1] W.E. Boyce and R.C. DiPrima, *Elementary Differential Equations*, Third Edition (New York: Wiley, 1976), Sections 7-6 through 7-9.

−2.44, −6.44, −6.56, and −10.56. Assured by this that the $\hat{x}_i$ exist, we go on to find

$$y_{lim} = -BA^{-1}x_0 = \begin{pmatrix} 0.827 \\ 0.173 \end{pmatrix}. \qquad \text{[Eq. A-10]}$$

That is, the probability that the Blue aircraft defeats the Red one is roughly 83 percent.

The eigenvalues of the plant matrix A have another use: they tell one whether or not it makes sense to look for the long-time limiting values of the absorbing boundary-state probabilities. Those probabilities are of interest if, but only if, an actual engagement can continue long enough for the system to be in boundary states with a probability near one. The time for this to happen is the time for which the transient states' probabilities are all much less than one.

When the eigenvalues of A are distinct (and, of course, have negative real parts) so that the $x_i(t)$ are linear combinations of the functions $\exp(\lambda_i t)$, that time will be no greater than the time at which $\exp(\lambda^* t)$ becomes much less than one, where λ^* is the eigenvalue closest to zero. The time to make that happen is approximately $2/\lambda^*$. If the rate parameters kb_1, … of our example are in inverse minutes, the mean time for the Blue aircraft to acquire a target is 20 seconds, the mean time for them to launch a successful missile is 15 seconds, and the mean time for them to break a Red lock is 30 seconds, while the mean time for the Reds to make lock is 60 seconds, the mean time for them to launch a successful missile is 30 seconds, and the mean time for them to break lock is 60 seconds. The negative eigenvalue of the plant matrix A with smallest magnitude is −2.44 inverse minutes, which implies that the time for the transient phase of the 1 vs. 1 engagement to be over is roughly 50 seconds. Very likely, the combatants will have enough fuel to fight that long, and so the long-time limit is meaningful in our example.

Although we have found numerical evaluation of Equation A-7 quite helpful in generating insight with examples of modest dimension, determining long-time limiting probabilities in this way may not be practical for systems of large dimension. In an important class of engagement models, however, the difficult task of computing $A^{-1} x_0$ can be done with a straightforward (although possibly lengthy) iterative scheme. That class is those engagement models whose diagrams are acyclic, like the example of Chapter 2. For these models, the plant matrix A is lower triangular, and this provides the iterative scheme. In these cases, one has

$$\begin{aligned} &a_{11}\hat{x}_1 = -1 \\ &a_{21}\hat{x}_1 + a_{22}\hat{x}_2 = 0 \qquad \text{[Eq. A-11]} \\ &\ldots \end{aligned}$$

so that

$$
\begin{aligned}
\hat{x}_1 &= -1/a_{11} \\
\hat{x}_2 &= -\hat{x}_1\, a_{21}/a_{22} \\
&\ldots
\end{aligned}
\qquad \text{[Eq. A-12]}
$$

With the $\hat{x}$ determined, the long-time limiting values of the absorbing boundary states follow from Equation A-7.

Determining the set of states actually occupied in a given engagement can be somewhat tedious. For example, finding the long-time limit of a missile-tracking engagement model for 4 vs. 4, when each aircraft has 6 missiles, involves more than 42,000 states. Obviously, it is not practical to evaluate these cases by hand. In these cases, we generally use C++ code to find both the set of states occupied and the solution (Equation A-12).

Appendix B

Analyzing Large State-Space Engagement Models with ASSIST and STEM

Evolution equations like Equations 2-5 and 2-6 of Chapter 2 are simple in principle. They are systems of linear ordinary differential equations with constant coefficients. The solutions of the initial value problems formed by such equations and specifications of the starting probabilities of the various states can be written symbolically—and perhaps treated practically— with matrix exponentiation, and they generally offer no difficulty to numerical solution by difference schemes, but they can be of large dimension.

A basic 4 vs. 4 engagement has 24 states and 32 transitions; adding 6 missiles to the Blue aircraft produces 1,632 states and 9,087 transitions; adding 6 missiles to both Red and Blue produces 108,534 states and 2,416,252 transitions. Also, it is somewhat awkward to determine the long-time limits of the absorbing boundary states by integrating the differential equations numerically. This appendix discusses NASA-developed tools that we have found helpful in dealing with both these issues.

The decision to apply the NASA tools to the problem of fighter combat was serendipitous. One of us was engaged in Markov analysis of fighter combat, and another of us was using the NASA tools for safety-related reliability analysis. The need for a tool to efficiently develop relatively complex fighter combat models surfaced during informal discussions. Using the NASA tools, we can develop and run complex models for Blue versus Red combat, including tracking of Blue missile use, in relatively short order. These results provide insight into the combat problem and support development of the SLAACM algorithms.

NASA has three tools that can be applied to fighter combat. Two are Markov computational analysis programs. The third is a sophisticated utility program that generates the inputs (models) for the analysis programs. The tools were developed by NASA to estimate failure probabilities in highly reliable, reconfigurable avionics and space electronics.

The two analysis programs are Scaled Taylor Exponential Matrix (STEM) [1] and Semi-Markov Unreliability Range Estimator (SURE).[2] The former is a pure Markov analysis tool in which all failure rates are constant with state probabilities

[1] NASA, *The PAWS and STEM Reliability Analysis Programs*, Technical Memorandum 100572, R. Butler and P. Stevenson, March 1988.

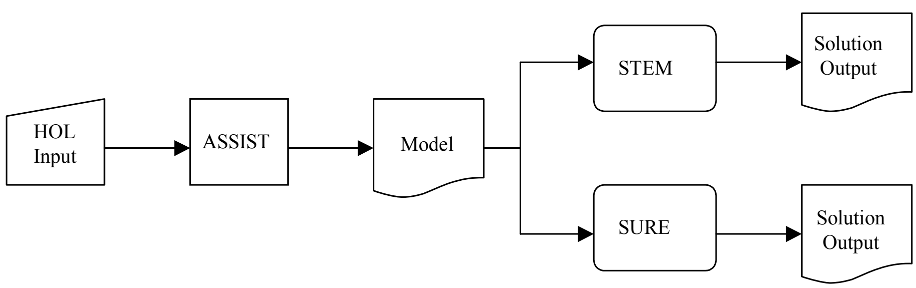
[2] NASA, *SURE Reliability Analysis, Program and Mathematics*, Technical Paper 2764, R. Butler and A. White, 1998.

having generalized exponential form. The latter is a semi-Markov model that allows use of non-exponential reconfiguration probabilities. STEM and SURE use identical input files, but some commands are processed only by SURE. For reasons discussed below, STEM is the tool used for fighter combat calculations. The utility program ASSIST (Abstract Semi-Markov Specification Interface to the SURE Tool)[3] generates the STEM and SURE input files, or "models." ASSIST allows the straightforward generation of extremely complex Markov models.[4]

Reliability analyses are typically conducted for short model times compared to the failure rates of the components studied, such as a 10-hour flight and a failure rate of 0.0001 failure per hour. For air combat models, we are interested in running the engagement to completion so we will run model time units on the order of the reciprocal of the lowest kill rate, e.g., 1,000 units for a kill rate of 0.001. STEM has no problem running such models, but SURE's mathematical algorithms that handle non-Markov recovery rates are not designed to run to completion. Consequently, the remainder of the discussion will focus on ASSIST and STEM.

All three programs were developed for UNIX platforms and have been converted to Windows. STEM is also available on LINUX. The programs are available at no charge from NASA Langley Research Center. Documentation includes an ASSIST users guide, a report on modeling techniques, and reports on SURE and STEM mathematics and performance. Figure B-1 shows the basic relationships of the tools.

Figure B-1. Basic Relationships of the Tools



HOL: Higher Order Language
ASSIST: Abstract Semi-Markov Specification Interface to the SURE Tool – Input model generator
Model: Designation for the input files for the Markov analysis programs, containing variables, states, transitions, and rates
STEM: Scaled Taylor Exponential Matrix – Markov analysis program
SURE: Semi-Markov Unreliability Range Estimator – Markov and Semi-Markov analysis program

FIGHTER COMBAT ANALYSIS

In this section, we use case examples of increasing complexity to describe the capabilities and limitations of the tools for analysis of fighter combat. All the analy-

[3] NASA Langley Research Center, *ASSIST User Manual*, S. Johnson and D. Boerschlein, September 1993.

[4] NASA, *Techniques for Modeling the Reliability of Fault-Tolerant Systems with the Markov State-Space Approach*, Reference Publication 1348, R. Butler and S. Johnson, September 1995.

ses use the common structure of an M vs. N, Blue vs. Red engagement with constant kill rates.

Example 1—Simple 2 vs. 2 Engagement

The 2 vs. 2 engagement demonstrates the basic structure of the models without generating extensive output. The Blue fighter kill rate is 0.1 per unit time, and the Red fighter kill rate is 0.01 per unit time. Listing 1 is the ASSIST code that generates the 2 vs. 2 model.

Listing 1. ASSIST Code for Generating the 2 vs. 2 Model

```
(* ASSIST model for Fighter Combat *)

LIST = 3; (* 1 lists accumulated death state probabilities only *)
(* 2 lists all state probabilities *)
(* 3 lists transitions and all state probabilities *)
ONEDEATH OFF; (* OFF enumerates all absorbing (Death) states *)
(* comment out to consolidate absorbing states *)

(* equipment list *)
nblue = 2; (* blue fighters *)
nred = 2; (* red fighters *)

(* kill probabilities *)
k_rate_blue = 0.1;
k_rate_red = 0.01;

SPACE = (blue: 0.nblue, red: 0.nred);
START = (nblue, nred);

DEATHIF (blue = 0) AND (red>0);
DEATHIF (red = 0) AND (blue>0);

(* Blue kills *)
IF (blue > 0)and (red>0) TRANTO red = red-1 BY blue*k_rate_blue;

(* Red kills *)
IF (red > 0 ) and (blue>0) TRANTO blue = blue-1 BY red*k_rate_red;
```

ASSIST uses a higher order definition language that allows algebraic manipulation of variables and compact description of Markov state transfers.[5] Several features are noteworthy in the listing above. The listing starts with the editing commands LIST and ONEDEATH that control SURE/STEM output. Next, the number of aircraft and their kill rates are input parameters defined as constant types. The SPACE statement defines the range of the Markov state space, i.e., in this case, the first state can vary from 0 to nblue where nblue equals 2. The START statement identifies the initial state populations. The DEATHIF statements define the absorbing states in the model; in this case, the two DEATHIF conditions represent Blue and Red wins. If the ONDEATH OFF command is enabled, STEM and SURE list all the individual absorbing states; otherwise, they combine the results for absorbing states into the appropriate Death states.[6]

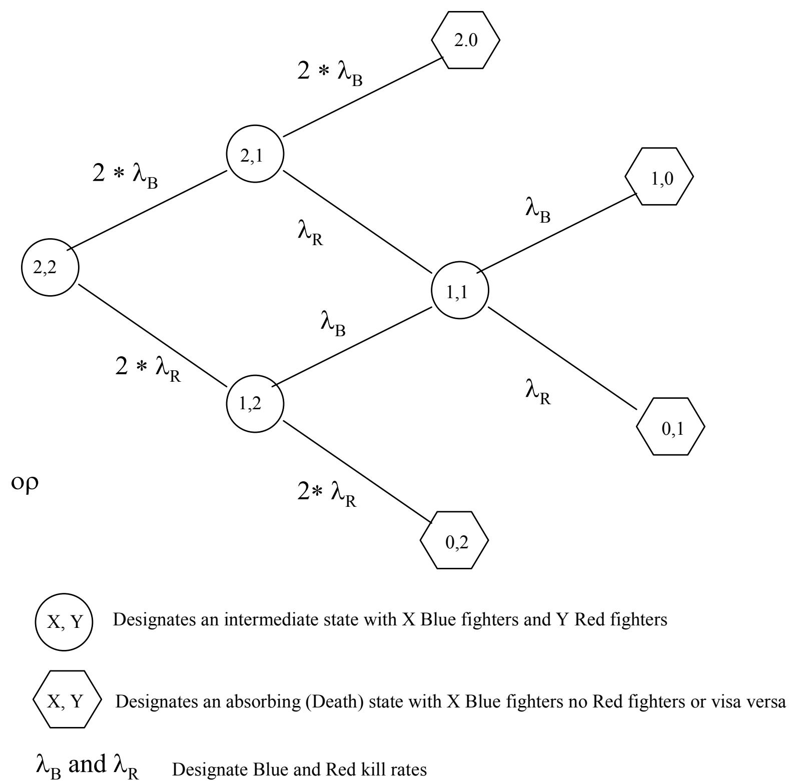
[5] ASSIST inputs are fully defined in the citations of Notes 1–3.

[6] SURE mathematics requires the existence of absorbing states, while STEM can handle models that include recovery from all states.

We now come to the powerful TRANTO statements, which define the conditions, nature, and rate of Markov state transfers. In this example, the transfers occur whenever both Blue and Red fighters are available; kills are made one at a time, and the kill rates are proportional to the number of killers. The two TRANTO statements above are adequate to generate the models for any combination of Red and Blue aircraft.

The state diagram for the 2 vs. 2 engagement, Figure B-2, is relatively simple. It shows that there are 8 total states, of which 4 are absorbing (Death) states and 8 are transitions.[7]

Figure B-2. 2 vs. 2 Engagement Diagram



Running ASSIST with the listing above generates the STEM/SURE input file or "model" shown in Listing 2. Note the 8 states, the 4 absorbing states (0, N or M, 0), and the 8 transitions. The file extension is ".mod."

[7] There is no (0, 0) absorbing state because we chose not to include the case in which the last two opponents simultaneously shoot each other down. Such a state could be modeled with an additional TRANTO statement, but determination of the controlling kill rate would take some careful thought. The pure statistical rate for independent events suggests that it would be equal to K_RATE_BLUE * K_RATE_RED, but simultaneous kills can happen only in certain conditions such as ramming, head-to-head gun attacks, and head-to-head missile attacks. These are only a subset of the configurations that make up the basic Blue and Red kill rates. We assessed the likely rate to be sufficiently low to justify ignoring the (0, 0) state until better data become available.

Listing 2. 2 vs. 2 Engagement Model

```
LIST = 3;
TIME = 100;
NBLUE = 2;
NRED = 2;
MIN_B = 0;
MIN_R = 0;
K_RATE_BLUE = 0.1;
K_RATE_RED = 0.01;


1(* 2,2 *), 2(* 2,1 *) = 2*K_RATE_BLUE;
1(* 2,2 *), 3(* 1,2 *) = 2*K_RATE_RED;
2(* 2,1 *), 4(* 2,0 *) = 2*K_RATE_BLUE;
2(* 2,1 *), 5(* 1,1 *) = 1*K_RATE_RED;
3(* 1,2 *), 5(* 1,1 *) = 1*K_RATE_BLUE;
3(* 1,2 *), 6(* 0,2 *) = 2*K_RATE_RED;
5(* 1,1 *), 7(* 1,0 *) = 1*K_RATE_BLUE;
5(* 1,1 *), 8(* 0,1 *) = 1*K_RATE_RED;

(* NUMBER OF STATES IN MODEL = 8 *)
(* NUMBER OF TRANSITIONS IN MODEL = 8 *)
```

The model file contains the input definitions and transition descriptions needed by SURE and STEM. Transitions are defined by source state number, destination state number, and rate of transfer between the source and destination. The state descriptions, e.g., (*2,2*), are included as comments; "(*" and "*)" are comment delimiters. Note that the absorbing states, such as state 4, (*2,0*), appear only in the destination column. The DEATHIF definitions and ONEDEATH command are not printed. Listing 3 is the STEM output file for the model above, run for 10 time units.

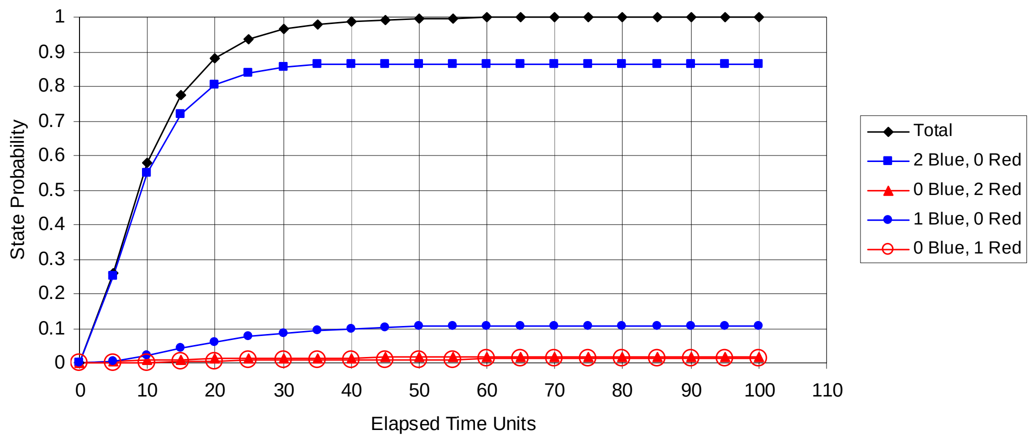
Listing 3. 2 vs. 2 Engagement STEM Output

```
Model = C:\Markov1\fighters\Report2x2.mod
——- RUN #1

D-STATE PROBABILITY ACCURACY
———-—————————-————
4 8.65800856430E-0001
6 1.51513103495E-0002
7 1.08204982318E-0001
8 1.08204982318E-0002

———-—————————-
TOTAL 9.99977647330E-0001 11 DIGITS

STATE PROBABILITY
———-——————————
1 2.78946809287E-0010
2 9.58618467009E-0009
3 1.22878668130E-0006
4 8.65800856430E-0001
5 2.11140182554E-0005
6 1.51513103495E-0002
7 1.08204982318E-0001
8 1.08204982318E-0002
```

Figure B-3 shows the STEM results for increasing units of combat time. As discussed in Chapter 2, we use the reciprocal of the lowest rate as a rule of thumb for

setting the model run time. In this case, the lowest rate is K_RATE_RED = 0.01 and its reciprocal is 100. By Time = 70, the total probability of being in an absorbing state is 0.9995, and at Time = 100, it is 0.99998, indicating that our rule of thumb is good. The state probabilities at Time = 100 for the individual absorbing states are (2B, 0R) = 0.8658, (1B, 0R) = 0.1082, (0B, 2R) = 0.0152, and (0B, 1R) = 0.0108.

Figure B-3. 2 vs. 2 Engagement Results versus Engagement Time



Example 2—2 vs. 2 Engagement with Blue Missiles

This example adds the tracking of Blue missiles to the simple 2 vs. 2 engagement. We assume that Blue aircraft each carry 6 missiles, each having a known single shot probability of kill, P_k. It is important to understand in this discussion that Blue and Red kills are still dependent only on the kill rate ratio, and that missile P_k is used only to calculate missile consumption. For simplicity and clarity, we assume, for this example, that Red aircraft have unlimited missiles.

Our basic approach is to expand the state space to (b0, b1, b2, b3, b4, b5, b6, r) to include separate states for Blue aircraft having 0 to 6 missiles. We still need only one state vector for Red aircraft. At the start of an engagement, we have 2 Blue aircraft in the b6 6-missile state and 2 Red aircraft in the r Red state.

We still transition from Red states in only two ways: Red can kill a Blue, or vice versa. Now, however, we can transition from a Blue state in three ways: Blue can kill a Red, Blue can fire and miss, or Red can kill a Blue. To formulate the transfer statements, we need to determine the missile miss rate. We know the Blue kill rate, λ, and we know the missile single shot probability of kill, P_k. We want to estimate the missile usage, including misses. We assume, for now, that the successful missile usage is one missile per kill, i.e., we do not fire salvos. We also assume the kill rate includes both the missile that hit and any that missed.

Now, we define μ to be the missile firing rate, and let *n* be the number of Blue aircraft engaged. The successful missile rate is then: $n*P_k*\mu$, and the unsuccessful missile rate (miss rate) is $n*(1-P_k)*\mu$.

Based on our assumptions, the successful missile rate must equal the kill rate, i.e.,

$$n*P_k*\mu = n*\lambda \qquad \text{[Eq. B-1]}$$

and, therefore, the missile firing rate is

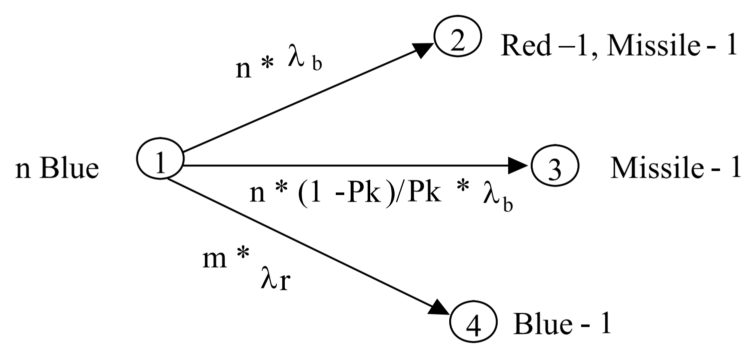
$$n*\mu = n*\frac{\lambda}{P_k}, \text{ and } \mu = \frac{\lambda}{P_k}. \qquad \text{[Eq. B-2]}$$

Substituting for μ, the miss rate, $n*(1-P_k)*\mu$, becomes

$$n*(1-P_k)*\frac{\lambda}{P_k} \text{ or } n*\frac{(1-P_k)}{P_k}*\lambda. \qquad \text{[Eq. B-3]}$$

Figure B-4 is the state transition diagram for the single missile per kill case.

Figure B-4. State Diagram for a Single Missile per Kill Engagement



Missile Firing Rate = μ
Missile Kill Rate = $P_k\mu$
Missile Miss Rate = $(1-P_k)\mu$
Engagement Kill Rates = λ_b, λ_r
Aircraft in Engagement = n Blue, m Red

We can generalize the derivation above in a straightforward manner for cases in which multiple missiles are fired per engagement by noting that the failure probability for *x* failures is $(1 - P_k)^x$ and substituting $[1 - (1-P_k)^x]$ for P_k:

- Define μ to be the missile firing rate.
- Let *n* be the number of Blue aircraft engaged.

- Let x be the number of missiles fired per kill.

The successful missile rate is $n*[1-(1-P_k)^x]*\mu$, and the unsuccessful missile rate (miss rate) is $n*(1-P_k)^x*\mu$.

Based on our assumptions, the successful missile rate must equal the kill rate, i.e.,

$$n*[1-(1-P_k)^x]*\mu = n*\lambda \qquad \text{[Eq. B-4]}$$

and, therefore, the missile firing rate is

$$n*\mu = n*\frac{\lambda}{\left[1-(1-P_k)^x\right]}, \text{ and } \mu = \frac{\lambda}{\left[1-(1-P_k)^x\right]}. \qquad \text{[Eq. B-5]}$$

The miss rate, $n*(1-P_k)^x*\mu$, now becomes

$$n*(1-P_k)^x*\frac{\lambda}{\left[1-(1-P_k)^x\right]} \text{ or } n*\frac{(1-P_k)^x}{\left[1-(1-P_k)^x\right]}*\lambda. \qquad \text{[Eq. B-6]}$$

For $x = 2$ missiles per kill, we have a miss rate of

$$n*\frac{1-2P_k+P_k^{\,2}}{2P_k-P_k^{\,2}}*\lambda. \qquad \text{[Eq. B-7]}$$

In addition to missile counting, we also add the capability for either Red or Blue to exit the combat when losses reach a preset level by assigning minimum values (min_b, min_r) to the DEATHIF conditions. If min_b = 0 and min_r = 0, the model corresponds to fighting to annihilation; higher values will generate models reflecting disengagement breakpoints. Listing 4 is the ASSIST code for the 2 vs. 2 case with missile tracking.

Listing 4. ASSIST Code for a 2 vs. 2 Engagement with Blue Missile Tracking

```
(* ASSIST code for 2 Blue vs. 2 Red with 6 missiles per Blue aircraft *)

LIST = 3; (* 1 lists accumulated death state probabilities only *)
(* 2 lists all state probabilities *)
(* 3 lists transitions and all state probabilities *)
ONEDEATH OFF; (* OFF enumerates all absorbing (Death) states *)
(* comment out to consolidate absorbing states *)

Time = 100;

(* equipment list *)
nblue = 2; (* blue fighters *)
nred = 2; (* red fighters *)

(* kill rates *)
k_rate_blue = 0.1; (* blue kill rate *)
k_rate_red = 0.01; (* red kill rate *)

pk = 0.85; (* missile kill probability *)
miss_rate = (1-pk)/pk*k_rate_blue; (* missile miss rate *)
```

```
(* minimum number of combatants for state pruning *)
min_b = 0; (* minimum number of blue aircraft *)
min_r = 0; (* minimum number of red aircraft *)

SPACE =
(n0:0.nblue,n1:0.nblue,n2:0.nblue,n3:0.nblue,n4:0.nblue,n5:0.nblue,n6:0.nblue,
red:0.nred);
START = (0,0,0,0,0,0,nblue,nred);

DEATHIF (n0+n1+n2+n3+n4+n5+n6)<=min_b;;
DEATHIF red <= min_r;

(* transition cases: Blue kill then Red kill*)
IF (n6>0) AND (red>0) THEN
TRANTO n6=n6-1, n5=n5+1, red=red-1 BY n6*k_rate_blue;
TRANTO n6=n6-1, n5=n5+1 BY n6*miss_rate;
TRANTO n6=n6-1 BY (n6/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

IF (n5>0) AND (red>0) THEN
TRANTO n5=n5-1, n4=n4+1, red=red-1 BY n5*k_rate_blue;
TRANTO n5=n5-1, n4=n4+1 BY n5*miss_rate;
TRANTO n5=n5-1 BY (n5/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

IF (n4>0) AND (red>0) THEN
TRANTO n4=n4-1, n3=n3+1, red=red-1 BY n4*k_rate_blue;
TRANTO n4=n4-1, n3=n3+1 BY n4*miss_rate;
TRANTO n4=n4-1 BY (n4/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

IF (n3>0) AND (red>0) THEN
TRANTO n3=n3-1, n2=n2+1, red=red-1 BY n3*k_rate_blue;
TRANTO n3=n3-1, n2=n2+1 BY n3*miss_rate;
TRANTO n3=n3-1 BY (n3/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

IF (n2>0) AND (red>0) THEN
TRANTO n2=n2-1, n1=n1+1, red=red-1 BY n2*k_rate_blue;
TRANTO n2=n2-1, n1=n1+1 BY n2*miss_rate;
TRANTO n2=n2-1 BY (n2/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

IF (n1>0) AND (red>0) THEN
TRANTO n1=n1-1, n0=n0+1, red=red-1 BY n1*k_rate_blue;
TRANTO n1=n1-1, n0=n0+1 BY n1*miss_rate;
TRANTO n1=n1-1 BY (n1/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;
ENDIF;

(* This line is commented out based on Blues ability to disengage after firing
all his missiles *)
(*IF (red>0) AND (n0>0) TRANTO n0=n0-1 BY
(n0/(n0+n1+n2+n3+n4+n5+n6))*red*k_rate_red;*)
```

Adding the missile Pk, missile miss rate, Blue and Red aircraft minimums, the expanded state space, and the starting conditions is straightforward. The expansion of the transfer statements deserves some discussion.

The basic transition logic is as follows:

- Separate sets of transfers are established for each Blue missile state.
- State transfers occur only when Blue aircraft are in the state and Red aircraft are available.

- Blue kills at a rate proportional to the total number of Blue aircraft. When Blue makes a kill, one aircraft is removed from the Blue state, one aircraft is added to the Blue state having one fewer missiles, and one aircraft is removed from the Red state.

- Blue misses at a rate proportional to the number of aircraft in the Blue state. When Blue misses, one aircraft is removed from the Blue state, one aircraft is added to the Blue state having one fewer missiles, but no aircraft are removed from the Red state.

- Red kills Blue at a rate proportional to the number of Red aircraft and the fraction of Blue aircraft in the Blue state. When Red makes a kill, one aircraft is removed from the Blue state.

- Only Red kills are possible when Blue has no missiles left. This TRANTO statement is commented out in the example listing using "(*" and "*)" operators based on the assumption that Blue can disengage at will. Another alternative would be to give Blues with no missiles a "guns only" kill rate.

Running the ASSIST program above generates a model having 103 states, including 34 absorbing (Death) states, and 301 transitions. Listing 5 shows the first and last few lines of the model file. The state notation in the model shows the state number and the contents of each element in the state. This model has eight elements in each state. The first seven elements are Blue aircraft with 0 to 6 missiles, and the last element is Red aircraft. Thus, State 1 is 1(0,0,0,0,0,0,2,2), indicating 2 Blues with 6 missiles and 2 Reds. The first transition is to 2(0,0,0,0,0,1,1,1), indicating that one Blue has fired a missile and killed one Red. A Blue absorbing state corresponds to 0s in all of the first seven elements, and a Red absorbing state has 0 in the eighth element.

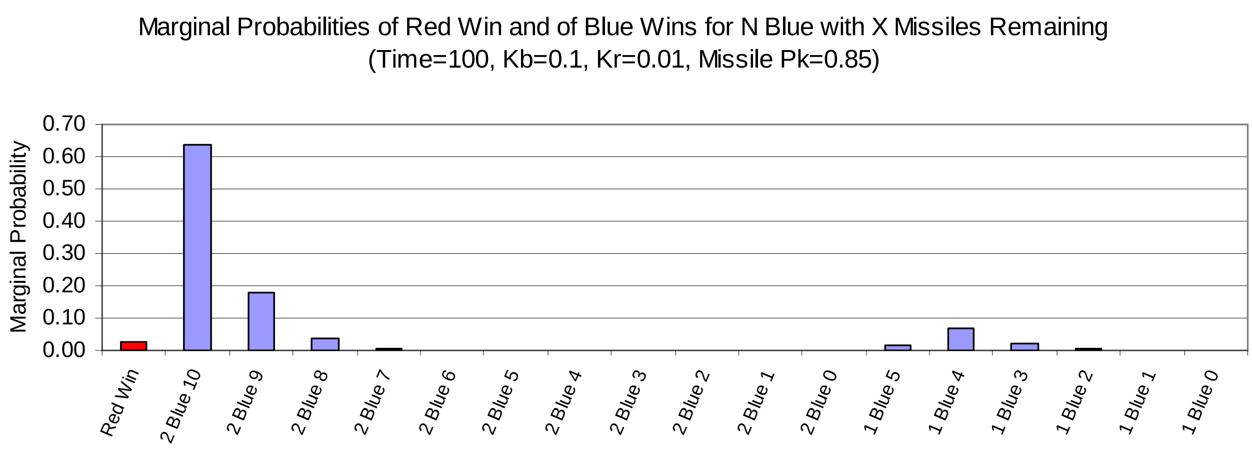
Listing 5. Model File Segments for 2 vs. 2 Engagement with Blue Missile Tracking

```
LIST = 3;
NBLUE = 2;
NRED = 2;
K_RATE_BLUE = 0.1;
K_RATE_RED = 0.01;
PK = 0.85;
MISS_RATE = (1-PK)/PK*K_RATE_BLUE;
MIN_B = 0;
MIN_R = 0;


1(* 0,0,0,0,0,0,2,2 *), 2(* 0,0,0,0,0,1,1,1 *) = 2*K_RATE_BLUE;
1(* 0,0,0,0,0,0,2,2 *), 3(* 0,0,0,0,0,1,1,2 *) = 2*MISS_RATE;
1(* 0,0,0,0,0,0,2,2 *), 4(* 0,0,0,0,0,0,1,2 *) = (2/(0+0+0+0+0+0+2))
*2*K_RATE_RED;
2(* 0,0,0,0,0,1,1,1 *), 5(* 0,0,0,0,0,2,0,0 *) = 1*K_RATE_BLUE;
2(* 0,0,0,0,0,1,1,1 *), 6(* 0,0,0,0,0,2,0,1 *) = 1*MISS_RATE;
2(* 0,0,0,0,0,1,1,1 *), 7(* 0,0,0,0,0,1,0,1 *) = (1/(0+0+0+0+0+1+1))
*1*K_RATE_RED;
2(* 0,0,0,0,0,1,1,1 *), 8(* 0,0,0,0,1,0,1,0 *) = 1*K_RATE_BLUE;
2(* 0,0,0,0,0,1,1,1 *), 9(* 0,0,0,0,1,0,1,1 *) = 1*MISS_RATE;
2(* 0,0,0,0,0,1,1,1 *), 10(* 0,0,0,0,0,0,1,1 *) = (1/(0+0+0+0+0+1+1))
```

```
*1*K_RATE_RED;
3(* 0,0,0,0,0,1,1,2 *), 6(* 0,0,0,0,0,2,0,1 *) = 1*K_RATE_BLUE;
3(* 0,0,0,0,0,1,1,2 *), 11(* 0,0,0,0,0,2,0,2 *) = 1*MISS_RATE;
3(* 0,0,0,0,0,1,1,2 *), 12(* 0,0,0,0,0,1,0,2 *) = (1/(0+0+0+0+0+1+1))
*2*K_RATE_RED;
3(* 0,0,0,0,0,1,1,2 *), 9(* 0,0,0,0,1,0,1,1 *) = 1*K_RATE_BLUE;
3(* 0,0,0,0,0,1,1,2 *), 13(* 0,0,0,0,1,0,1,2 *) = 1*MISS_RATE;
3(* 0,0,0,0,0,1,1,2 *), 4(* 0,0,0,0,0,0,1,2 *) = (1/(0+0+0+0+0+1+1))
                                        .
                                        .
                                        .
97(* 1,0,1,0,0,0,0,2 *), 99(* 1,1,0,0,0,0,0,1 *) = 1*K_RATE_BLUE;
97(* 1,0,1,0,0,0,0,2 *), 100(* 1,1,0,0,0,0,0,2 *) = 1*MISS_RATE;
97(* 1,0,1,0,0,0,0,2 *), 76(* 1,0,0,0,0,0,0,2 *) = (1/(1+0+1+0+0+0+0))
*2*K_RATE_RED;
99(* 1,1,0,0,0,0,0,1 *), 101(* 2,0,0,0,0,0,0,0 *) = 1*K_RATE_BLUE;
99(* 1,1,0,0,0,0,0,1 *), 102(* 2,0,0,0,0,0,0,1 *) = 1*MISS_RATE;
99(* 1,1,0,0,0,0,0,1 *), 72(* 1,0,0,0,0,0,0,1 *) = (1/(1+1+0+0+0+0+0))
*1*K_RATE_RED;
100(* 1,1,0,0,0,0,0,2 *), 102(* 2,0,0,0,0,0,0,1 *) = 1*K_RATE_BLUE;
100(* 1,1,0,0,0,0,0,2 *), 103(* 2,0,0,0,0,0,0,2 *) = 1*MISS_RATE;
100(* 1,1,0,0,0,0,0,2 *), 76(* 1,0,0,0,0,0,0,2 *) = (1/(1+1+0+0+0+0+0))
*2*K_RATE_RED;

(* NUMBER OF STATES IN MODEL = 103 *)
(* NUMBER OF TRANSITIONS IN MODEL = 285 *)
```

At 100 time units, the engagement is 0.99994 complete; the sums of the absorbing missile tracking states agree within four significant figures with the sums of the absorbing states for the simple 2 vs. 2 example. Figure B-5 combines the probabilities for the absorbing states to show the marginal probabilities of Red victory and the marginal probabilities of Blue victory with specific numbers of aircraft and missiles after a 2 vs. 2 engagement of 100 time units.

Figure B-5. 2 vs. 2 Engagement Aircraft and Missile Configuration Probabilities



Example 3—4 vs. 4+4+4 Sequential Engagement

The standard SLAACM campaign scenario has 4 Blue defenders engaging 4 Red long-range escorts (LEs), 4 close escorts (CEs), and 4 bombers sequentially. This sequential engagement can be modeled using ASSIST by imposing on the TRANTO statements the conditions that Blue cannot engage the close escorts or bombers until all the LEs are dead and cannot engage the bombers until all the

CEs are also dead. We include individual kill rates for the three pairs of combatants. Listing 6 is the ASSIST code for the sequential engagement. The resulting model has 38 states and 48 transitions when the Blue breakpoint is 2 losses (min_b = 2 in the listing), and has 64 states and 96 transitions when Blue fights to annihilation (min_b = 0). Figure B-6 shows the marginal probabilities for the outcomes for the 2 Blue breakpoint, and Figure B-7 shows the corresponding expected number of surviving aircraft types.

Listing 6. 4 Blue Sequentially Engaging 4 Red LE, 4 Red CE, and 4 Red Bombers

```
(* ASSIST model for Fighter Combat *)

(* This is multistage combat where blue aircraft sequentially engage*)
(* three classes of red aircraft *)
(* This version does not include missiles *)

LIST = 3; (* 1 lists accumulated death state probabilities only *)
(* 2 lists all state probabilities *)
(* 3 lists transitions and all state probabilities *)
ONEDEATH OFF; (* OFF enumerates all absorbing (Death) states *)
(* comment out to consolidate absorbing states *)

(*”Time = 5 to 100 By 5;”*)
(* the time is set here - no matter what the STEM GUI shows *)
Time = 1000;

(* equipment list *)
nblue = 4; (* blue fighters *)
nred1 = 4; (* red fighters - lead escorts *)
nred2 = 4; (* red fighters - close escorts *)
nred3 = 4; (* red bombers or fighter bombers *)

min_b = 2;
min_r1 = 0;
min_r2 = 0;
min_r3 = 0;

(* kill probabilities *)
(* Note: the 1st 12 characters of a variable must be unique *)
blue_red1_k_rate = 0.1; (* 0.1 *)
blue_red2_k_rate = 0.1; (* 0.2 *)
blue_red3_k_rate = 0.1; (* 0.4 *)
k_rate_red1 = 0.02; (* 0.05 *)
k_rate_red2 = 0.01; (* 0.01 *)
k_rate_red3 = 0.005; (* 0.005 *)

SPACE = (blue: 0.nblue, red1: 0.nred1, red2: 0.nred2, red3: 0.nred3);
START = (nblue, nred1, nred2, nred3);

DEATHIF (blue = min_b) AND ((red1>0) OR (red2>0) OR (red3>0));
DEATHIF (red1 = min_r1) AND (red2 = min_r2) AND (red3 = min_r3) AND (blue>0);

(* Phase 1 combat *)
(* Blue kills *)
IF (blue > min_b) and (red1 > min_r1) TRANTO red1 = red1-1 BY
blue*blue_red1_k_rate;
(* Red kills *)
IF (red1 > min_r1 ) and (blue>min_b) TRANTO blue = blue-1 BY red1*k_rate_red1;

(* Phase 2 combat *)
(* Blue kills *)
IF (blue > min_b) and (red1=min_r1) and (red2 > min_r2) TRANTO red2 = red2-1
BY blue*blue_red2_k_rate;
(* Red kills *)
```

```
IF (red2 > min_r2 ) and (red1=min_r1) and (blue>min_b) TRANTO blue = blue-1 BY
red2*k_rate_red2;

(* Phase 3 combat *)
(* Blue kills *)
IF (blue > min_b) and (red3 > min_r3) TRANTO red3 = red3-1 BY
blue*blue_red3_k_rate;
(* Red kills *)
IF (red3 > min_r3 ) and (red1=min_r1) and (red2=min_r2) and (blue>min_b)
TRANTO blue = blue-1 BY red3*k_rate_red3;
```

Figure B-6. Sequential Engagement Marginal Probabilities

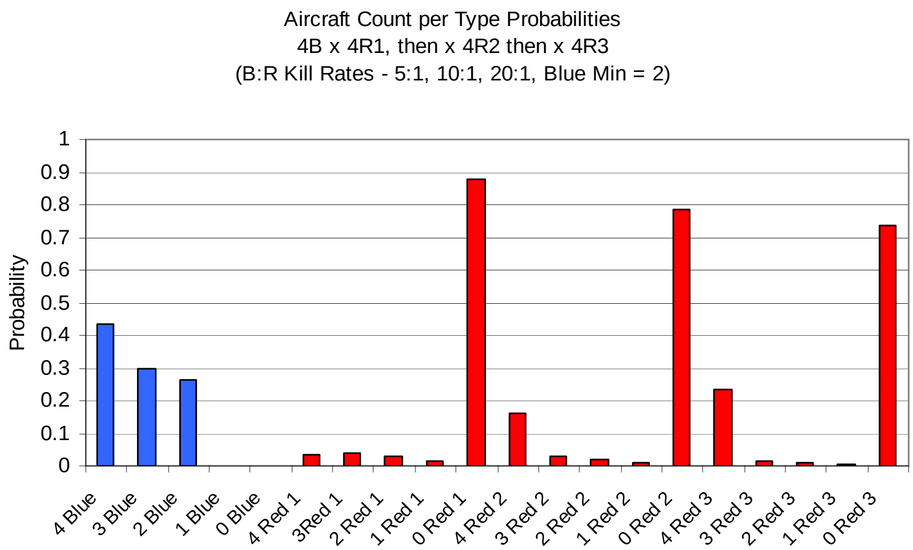
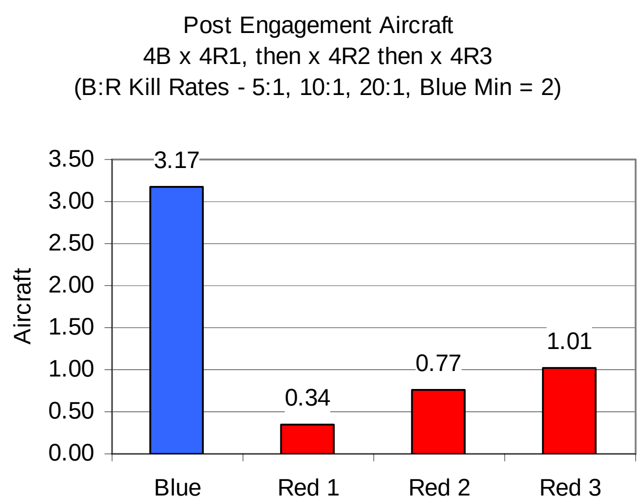


Figure B-7. Sequential Engagement Surviving Aircraft



Example 4—4 vs. 4+4+4 Sequential Engagement with Missiles

The last example is the standard sequential engagement with missiles on the Blue aircraft. Listing 7 is the ASSIST code for this case. This listing generates a model having 3,695 states and 21,729 transitions for the 2 Blue breakpoint case and 3,823 states and 23,454 transitions for the Blue fight to annihilation case. Figure

B-8 shows the marginal probabilities of missile counts combined with information on how many aircraft are carrying the missiles for the case in which Blue breaks after two losses. Figure B-9 shows the remaining aircraft count by type. Table B-1 contains the summary results for both the 2 Blue breakpoint and Blue fight to annihilation cases.

Listing 7. 4 Blue vs. 4 Red LE, 4 Red CE, and 4 Bombers Sequentially with Blue Missiles

```
(* ASSIST model for *)
(* This version includes sequential combat with 3 types of red fighters *)
(* and includes Blue missiles *)
(* Fighter Combat using Dave Lee’s construct for blue missile counting*)
(* 7/20/05 This version includes missiles for blues *)
(* 7/20/05 THIS VERSION INCLUDES 6 BLUE MISSILES
(* THIS VERSION CAN LIMIT STATES BASED ON A MINIMUM NUMBER OF COMBATANTS *)
(* The kill rate is for one-on-one and is multiplied for N killers *)
(* Assume one missile per kill even with N killers *)
(* Therefore, the kill rate includes one hit and all misses *)
(* Find miss rate based on ratio of Pmiss:Pk *)

LIST = 3; (* 1 lists accumulated death state probabilities only *)
(* 2 lists all state probabilities *)
(* 3 lists transitions and all state probabilities *)
ONEDEATH OFF; (* OFF enumerates all absorbing (Death) states *)
(* comment out to consolidate absorbing states *)

(*”Time=10 TO+ 100 BY 10;”*)
Time = 1000;

(* equipment list *)
nblue = 4; (* blue fighters *)
nred1 = 4; (* red fighters *)
nred2 = 4;
nred3 = 4;


(* missile equipage *)
b_msls = 6; (* blue missiles *)

(* the first 12 characters in a variable name are significant and must be
unique *)

(* kill rates *)
red1_k_rate_blue = 0.1; (* B:R1 = 5:1 *)
red2_k_rate_blue = 0.1; (* B:R2 = 10:1 *)
red3_k_rate_blue = 0.1; (* B:R3 = 20:1 *)
k_rate_red1 = 0.02;
k_rate_red2 = 0.01;
k_rate_red3 = 0.005;


(* blue missiles miss rates *)
bpk = 0.85; (* missile kill probability *)
red1_bmiss_rate = (1-bpk)/bpk*red1_k_rate_blue; (* missile miss rate against
red1 *)
red2_bmiss_rate = (1-bpk)/bpk*red2_k_rate_blue; (* missile miss rate against
red2 *)
red3_bmiss_rate = (1-bpk)/bpk*red3_k_rate_blue; (* missile miss rate against
red3 *)

(* minimum number of combatants for state pruning *)
min_b = 2; (* minimum number of blue aircraft *)
min_r1 = 0; (* minimum number of red1 aircraft *)
min_r2 = 0;
min_r3 = 0;
```

```
SPACE = (b:array[0.b_msls]of 0.nblue, red1: 0.nred1, red2: 0.nred2, red3:
0.nred3);

(* states 0 to max missiles *)
(* THIS STATEMENT MUST BE CHANGED WHEN THE MISSILE COUNT IS CHANGED! *)
START = (0,0,0,0,0,0,nblue,nred1,nred2,nred3);

DEATHIF (sum(b)<= min_b) OR (b[0]=4);
DEATHIF (red1 = min_r1) AND (red2 = min_r2) AND (red3 = min_r3);

(* state transitions and kills *)
(* advanced escorts *)
FOR I IN [1.b_msls]
IF (b[I]>0) AND (sum(b)>min_b) AND (red1>min_r1) THEN
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1, red1 =red1-1 BY b[I]*red1_k_rate_blue;
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1 BY b[I]*red1_bmiss_rate;
TRANTO b[I]=b[I]-1 BY (b[I]/sum(b))*red1*k_rate_red1;
ENDIF;
ENDFOR;
(* no missile case against advanced escorts *)
IF (b[0]>0) AND (sum(b)>min_b) AND (red1 > min_r1) THEN
(* TRANTO red1=red1-1 BY b[0]*red1_k_rate_blue; *) (* NO BLUE KILLS WITHOUT
MISSILES *)
(* TRANTO b[0]=b[0]-1 BY (b[0]/sum(b))*red1*k_rate_red1; *) (* BLUE ESCAPES IF
NO MISSILES *)
ENDIF;

(* close escorts *)
IF red1 = min_r1 THEN
FOR I IN [1.b_msls]
IF (b[I]>0) AND (sum(b)>min_b) AND (red2 > min_r2) THEN
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1, red2 =red2-1 BY b[I]*red2_k_rate_blue;
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1 BY b[I]*red2_bmiss_rate;
TRANTO b[I]=b[I]-1 BY (b[I]/sum(b))*red2*k_rate_red2;
ENDIF;
ENDFOR;
(* no missile case against close escorts *)
IF (b[0]>0) AND (sum(b)>min_b) AND (red2 > min_r2) THEN
(* TRANTO red2=red2-1 BY b[0]*red2_k_rate_blue; *) (* NO BLUE KILLS WITHOUT
MISSILES *)
(* TRANTO b[0]=b[0]-1 BY (b[0]/sum(b))*red2*k_rate_red2; *) (* BLUE ESCAPES IF
NO MISSILES *)
ENDIF;
ENDIF;

(* bombers *)
IF (red2 = min_r2) AND (red1 = min_r1) THEN
FOR I IN [1.b_msls]
IF (b[I]>0) AND (sum(b)>min_b) AND (red3 > min_r3) THEN
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1, red3 =red3-1 BY b[I]*red3_k_rate_blue;
TRANTO b[I]=b[I]-1, b[I-1]=b[I-1]+1 BY b[I]*red3_bmiss_rate;
TRANTO b[I]=b[I]-1 BY (b[I]/sum(b))*red3*k_rate_red3;
ENDIF;
ENDFOR;
(* no missiles case against bombers *)
IF (b[0]>0) AND (sum(b)>min_b) AND (red3 > min_r3)THEN
(* TRANTO red3=red3-1 BY b[0]*red3_k_rate_blue; *) (* NO BLUE KILLS WITHOUT
MISSILES *)
(* TRANTO b[0]=b[0]-1 BY (b[0]/sum(b))*red3*k_rate_red3; *) (* BLUES ESCAPES
IF NO MISSILES *)
ENDIF;
ENDIF;

(* FOR NO MISSILE CASES: *)
(* use only red rates if blue has no missiles left and cannot escape *)
(*—- comment out all statements if blue can escape *)
(*—- include blue kill statements to continue combat without missiles *)
(* possibly add rates above and TRANTO statments here for Blue guns *)
```

Figure B-8. Sequential Engagement Missile Marginal Probabilities with Carrier Aircraft Information

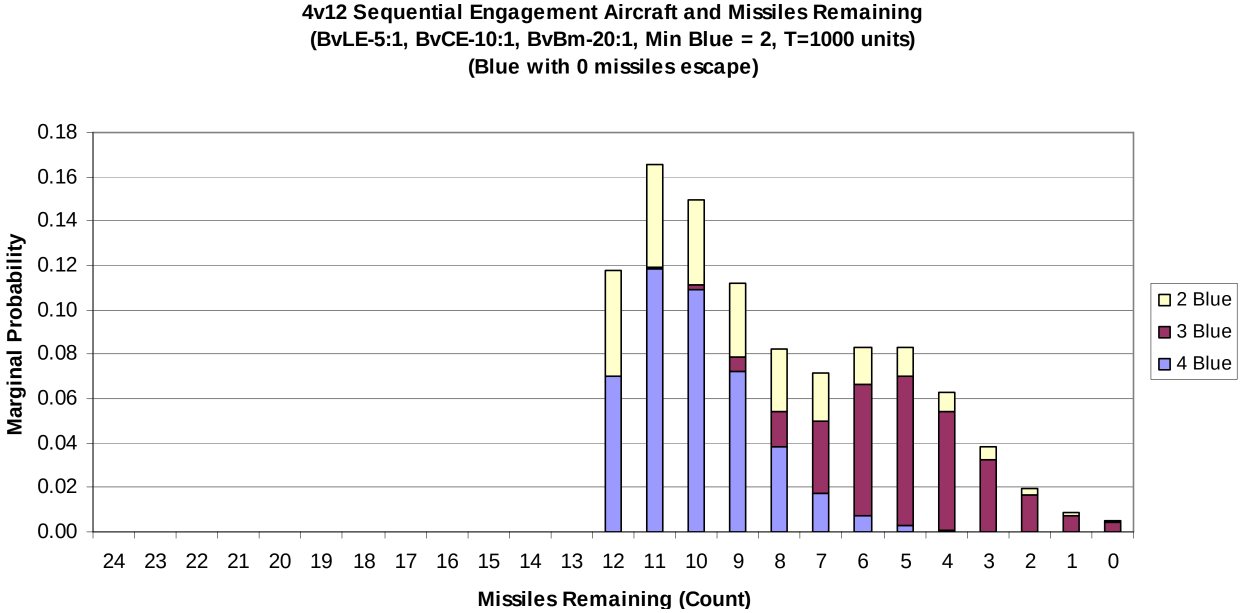


Figure B-9. Sequential Engagement with Missiles, Remaining Aircraft

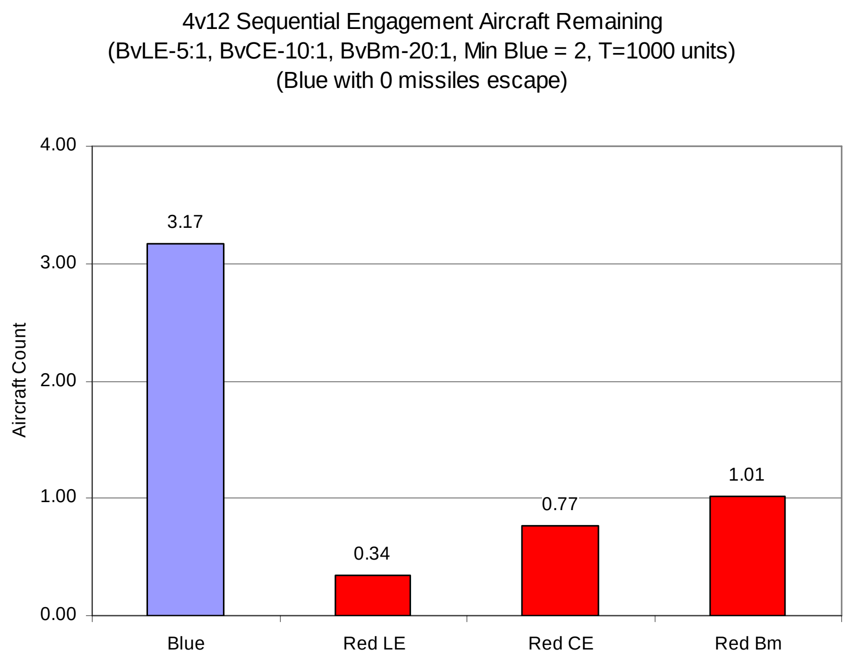


Table B-1. Summary Engagement Results

Scenario	Blue remaining	Red remaining	Missiles used
2 Blue breakpoint	3.2	2.1	16
Blue fight to annihilation	3.0	0.8	18

SUMMARY

In this appendix, we have shown how the NASA-developed Markov tools ASSIST and STEM can be used for analysis of increasingly complex air combat engagements. We have shown through examples how ASSIST is particularly useful for supporting parametric analysis involving large state-space problems. The NASA tools have proved useful for standalone analyses, for prototyping of SLAACM engagements, and for independent confirmation of SLAAM calculations.

Appendix C
Alternatives to Computational Probabilistic Tools

In this appendix, we compare results of two familiar models, the Lanchester force-on-force model[1] and simulation modeling, with the results of the probabilistic engagement model applied to some specific problems.

LANCHESTER FORCE-ON-FORCE MODEL

One alternative to the probabilistic engagement model is the classic Lanchester force-on-force model. In this engagement model, the time rate of change of the number of members of a side is proportional to the number of members of the other side. This leads to the initial value problem Equation C-1.

$$\begin{aligned} \dot{n}_b &= -k_r n_r \\ \dot{n}_r &= -k_b n_b \\ n_b(0) = b_0; &\quad n_r(0) = r_0 \end{aligned} \qquad . \qquad \text{[Eq. C-1]}$$

The differential equations in Equation C-1 can be solved readily in closed form with exponential functions.[2]

The Lanchester force-on-force model is, of course, entirely deterministic. In fact, its outcomes are determined by values of a certain ratio, called the strength ratio. To see this, we may multiply the first differential equation by $k_b n_b$ and the second by $k_r n_r$, subtract the second product from the first, and integrate. This leads to

$$k_b(b_0^2 - n_b^2) = k_r(r_0^2 - n_r^2). \qquad \text{[Eq. C-2]}$$

Using Equation C-2, we can develop the strength ratio, S, which is defined by Equation C-3.

$$S \equiv \frac{k_b(b_0^2 - b_{bk}^2)}{k_r(r_0^2 - r_{bk}^2)}. \qquad \text{[Eq. C-3]}$$

Examining Equation C-3, one sees that if Red breaks when they have r_{bk} members, and Blue breaks when they have b_{bk}, then Blue will win in finite time when

---

[1] James G. Taylor, *Lanchester Models of Warfare*, Volume I (Operations Research Society of America, March 1983), p 83.

[2] See Note 1.

the strength ratio S is greater than one. If $S < 1$, Red wins in finite time. If $S = 1$ and at least one of r_{bk} and b_{bk} is positive, the engagement ends at finite time in a tie; otherwise, the engagement continues indefinitely, with n_b and n_r tending to zero. The fact that the effectiveness parameters k_b and k_r appear linearly in the strength ratio S while the sides' numbers appear quadratically is called the "Lanchester square law." That law has been used to argue, not always convincingly, that "quantity trumps quality" for the strength of military forces.

If the Blue side wins the engagement, its losses are

$$b_0 - b = b_0 - \sqrt{b_0^2 - \frac{k_r(r_0^2 - r_{bk}^2)}{k_b b_0^2}}\,. \qquad \text{[Eq. C-4]}$$

Let us compare the results of analyzing a specific M vs. N engagement with the Lanchester force-on-force model and with the probabilistic engagement model. For a 4 vs. 8 engagement in which $\kappa = 10$, both sides fighting to annihilation, the Lanchester model predicts that the Blue side will win, losing 0.902 aircraft (interpreting fractional aircraft losses is one difficulty of using the Lanchester model). The Reds lose 8 aircraft.

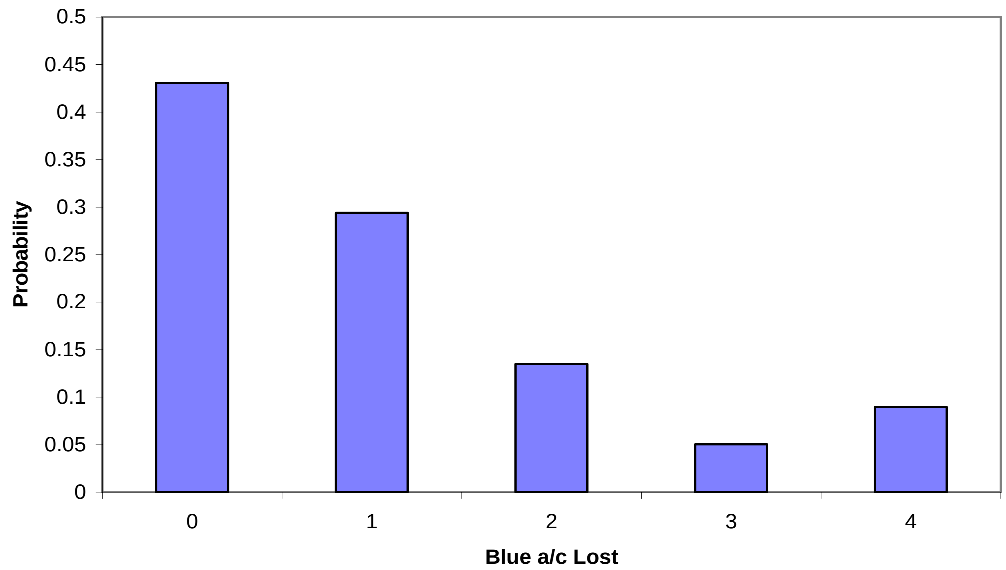
The M vs. N probabilistic engagement model predicts that Blue will win with probability 0.91, with the expected value of the Blue losses equal to 1.075 and the expected value of the Red losses equal to 7.595.

The expected loss values are not too different from those of the Lanchester result. Nevertheless, the information that Blue's victory is somewhat in doubt—Red has nearly a 10 percent chance of winning—may be important information for planners to have.

The distribution of Blue losses, including the probabilities of losing 0, 1, 2, 3, or 4 Blue aircraft, shown in Figure C-1, is also informative. There is a 43 percent probability of losing no aircraft, a 29 percent probability of losing 1 aircraft, and a 28 percent probability of losing more than 2 aircraft. Thus there is a substantial probability of losses almost twice the expected value of 1.075. This sort of information may be particularly useful to loss-averse planners.

Certainly, there are more sophisticated Lanchester models than the one considered here. There are also more sophisticated computational probabilistic models than the one considered here. We believe that the comparison in this section does, however, indicate the relative helpfulness of the two classes of models to military analysts.

Figure C-1. Distribution of Blue Losses



SIMULATION MODELING

Simulation models have significant strengths. They enable consideration of many factors and, like probabilistic engagement models, produce outcome statistics that give planners insight into important uncertainties. However, simulation models also have defects directly corresponding to their virtues. Setting up and operating large-scale simulation models can take significant time, making them less helpful to planners who need to consider wide varieties of cases quickly. Moreover, with many inputs, it can be hard to tell just what inputs are responsible for which output effects.

Simulation models require careful consideration of the numbers of runs required to establish confidence in certain results, and they make significant demands on random number generation. Let us illustrate these points with an example. We consider the engagement for $\kappa = 3$. Table C-1 shows the probabilities of the four outcome states, readily computed with the iterative method of Appendix A for the probabilistic model.

Table C-1. Outcome State Probabilities

Outcome state	Probability
(4, 0)	0.6000
(3, 0)	0.2423
(0, 3)	0.0808
(0, 4)	0.0769

Figures C-2 and C-3 show results for the probability of state (4,0) and for the probability of state (0,4), respectively, obtained with 100-run simulations.

Figure C-2. P(4, 0) from 100-Run Simulations

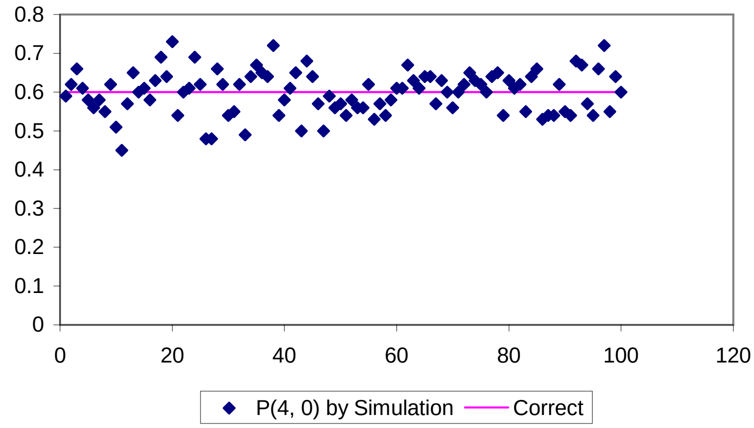
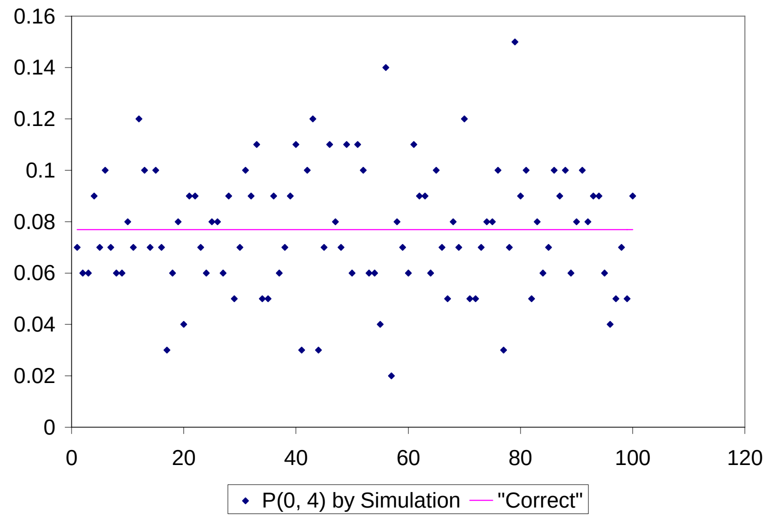


Figure C-3. P(0, 4) from 100-Run Simulations

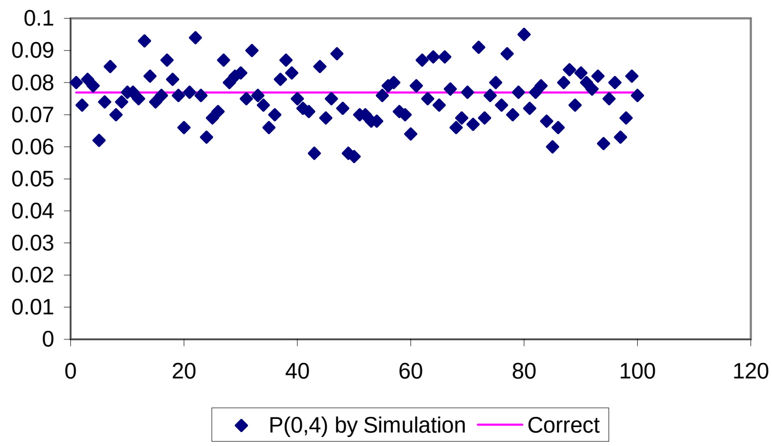


The value of P(4, 0) is not determined very well with 100-run simulations, and the value of the rarer outcome, (0, 4), is determined badly. These results for such a simple case may serve as a warning to users of simulations who pick a number of runs casually. ("Oh, a hundred runs ought to be enough.")

Either analysis[3] or experiment show that at least 1,000 runs are required to bring below 11 percent the coefficient of determination of values of P(0, 4) inferred by simulation. Figure C-4 shows some example results.

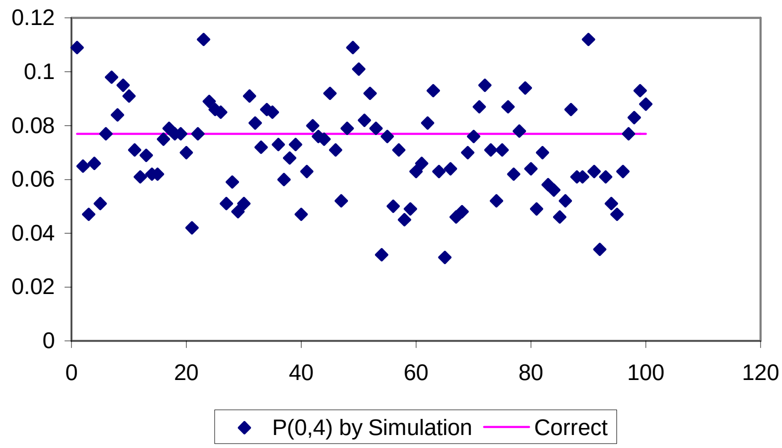
[3] Reference for analytic determination of the number of runs required.

Figure C-4. P(0, 4) from 1,000-Run Simulations



The results shown in Figures C-2 through C-4 were obtained with a random number generator that we believe to be a good one.[4] Careless use of the random number generator in a popular application gave the results shown in Figure C-5

Figure C-5. P(0, 4) from 1,000-Run Simulations
with Poor Random Number Generator



Simulations made with care are thoroughly useful. SLAACM is by no means intended to compete with well-thought-out, large-scale simulations. Rather, we believe it is a useful complement to such models, valuable because it allows rapid analysis of several cases in the PC environment.

[4] It is the routine "ran1" described by W.H. Press and others in *Numerical Methods in C*, Second Edition (Cambridge University Press, 1996), p. 280.

Appendix D
Advanced Probabilistic Engagement Models

The simple M vs. N probabilistic engagement model offers an organized approach to modeling the statistics of air-to-air engagements. In addition, when the engagements go to completion (that is, to the long-time limit), it is straightforward to obtain its one parameter, the kill-rate ratio, κ, from observations of loss ratios obtained from combat data, training exercises, and simulations. However, the simple model has limitations. Therefore, in this appendix, we discuss more sophisticated models that eliminate some of those limitations. Straightforward modifications to SLAACM will allow any of these models to be used in place of the M vs. N probabilistic engagement model.

DETAILED ENGAGEMENT MANAGEMENT

One limitation of the M vs. N probabilistic engagement model is the assumption of independent events. After all, the Blue and Red aircraft are attacking fixed sets of airplanes, and, particularly as numbers grow small, the independence may seem unrealistic.

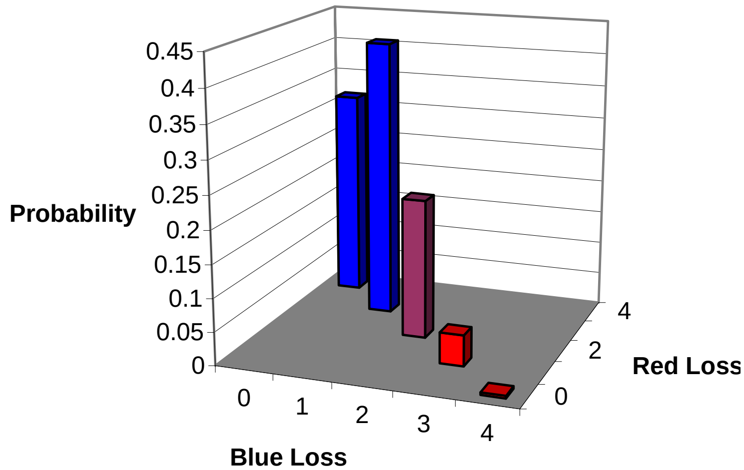
One way to remove this limitation is to develop models with more specific information about the sides' management of the engagement. For example, if a 4 vs. 4 engagement splits immediately into four 1 vs. 1 engagements, the independence assumption is more realistic.

Suppose that both sides' battle management has the 4 vs. 4 engagement split into four 1 vs. 1 engagements, each of which goes to the long-time limit, and that afterwards the Red and Blue survivors depart without further engagement. Then there are five outcome states, and their probability distribution is binomial. Specifically,

$$P(m,n) = B(m-n,\, 4,\, \kappa/(\kappa+1)) . \qquad \text{[Eq. D-1]}$$

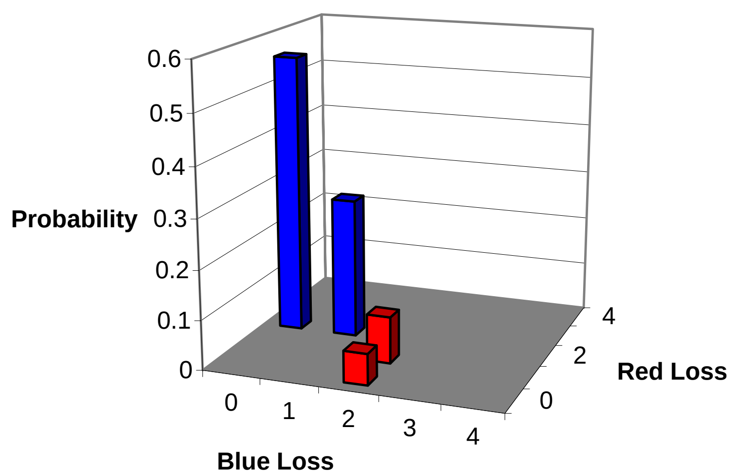
Equation D-1 also holds when the time-to-kill distributions of the two sides are not exponential; in such cases, the quantity $\kappa/(\kappa + 1)$ is replaced by the probability that the Blue side makes the first kill in a 1 vs. 1 engagement. Figure D-1 shows the outcome loss distribution when $\kappa = 3$.

Figure D-1. Loss Distribution: 4 vs. 4 Engagement
Becomes Four 1 vs. 1 Engagements



If, after the 4 vs. 4 engagement breaks into four 1 vs. 1 engagements, each side breaks away from the entire engagement upon suffering two losses, the engagement statistics are binary through the first two kills. If these result either in two Blue losses or in two Red losses, the engagement ends. Otherwise, the first two kills have left three aircraft on each side, and the engagement ends with the next kill. Figure D-2 shows the loss distribution for this battle management scenario, again for $\kappa = 3$.

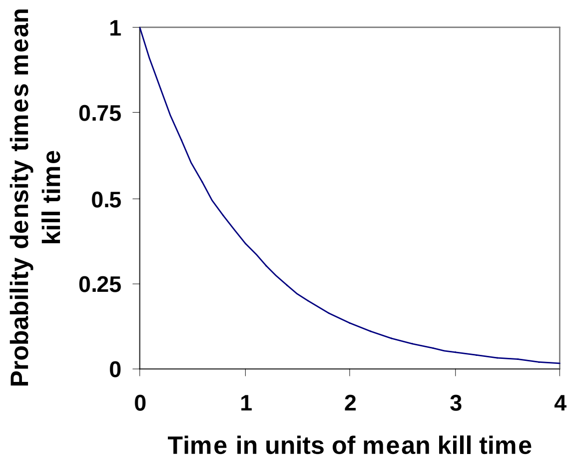
Figure D-2. Loss Distribution: Alternate Battle Management



MULTIPHASE KILL PROCESSES

Another concern about the simple M vs. N probabilistic engagement model is that the modes of the assumed exponential time-between-kill distributions are at zero (Figure D-3). It may seem unrealistic to have zero as the most likely time between kills.

Figure D-3. Exponential Distribution

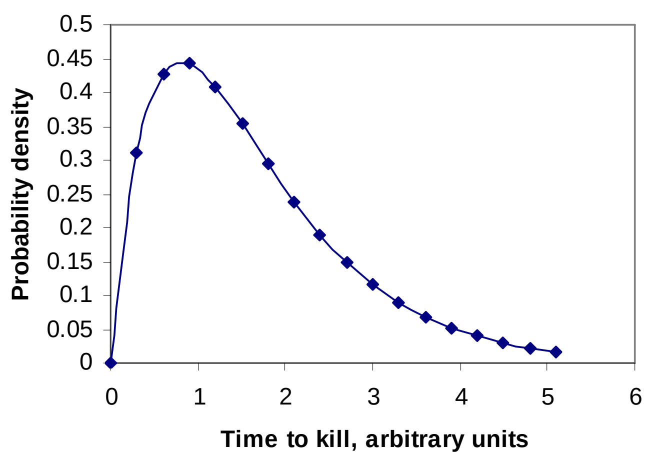


This concern may be addressed by modeling the kill process as taking place in two phases. In the first phase, a combatant seeks opponents. This phase ends when the combatant's fire control system locks onto an opponent.

The second phase begins with lock-on and ends when the combatant has fired a successful missile. The assumption of exponential distributions of the times required for these two phases is not as heavy as the assumption of an exponential distribution: with effective guidance from C^3ISR systems, a combatant's lock-on might occur essentially immediately after the search begins, and, with a firm lock and an effective missile, firing of a successful missile might occur essentially immediately after lock-on.

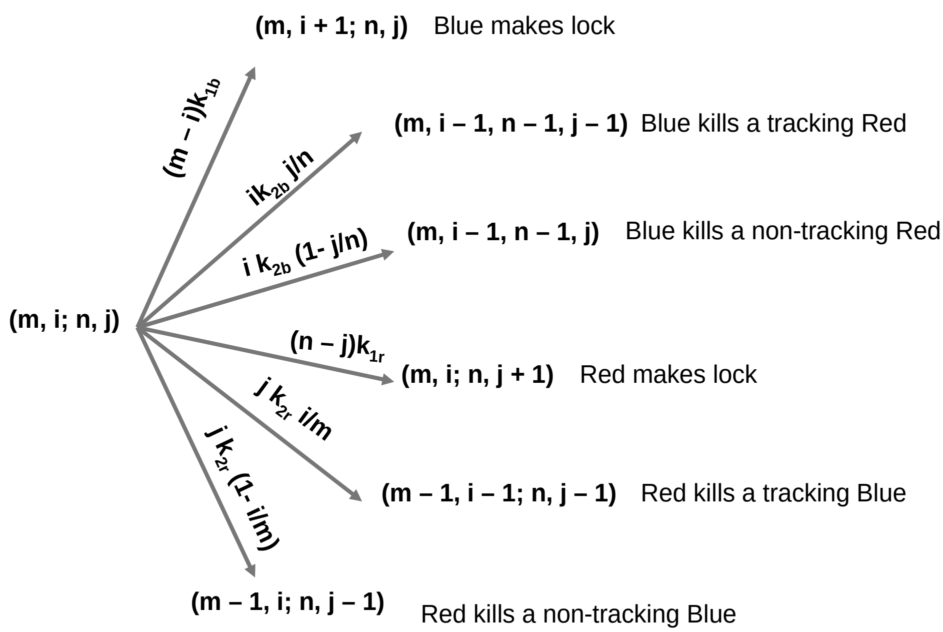
The total time-between-kills distribution for a two-phase process is the convolution of two exponentials. (It is not always the Erlang distribution; that happens only when the parameters of the two exponential distributions are the same.) Figure D-4 shows an example.

Figure D-4. Time-between-Kills Distribution for Two-Phase Kill Process



The engagement state with two-phase kill processes requires more information to describe it than the two numbers required when single-phase kills were used. We use the state variable (m, i; n, j), where m is the number of Blue aircraft in the engagement, i is the number of Blues that have locked onto targets, n is the number of Red aircraft in the engagement, and j is the number of Reds that have locked onto targets. Instead of the two-valued state transition diagram of Figure D-1, the state transition diagram for the two-phase kill process has six elements, as shown in Figure D-5. Text on the arrows in the figure shows the rates for the respective transitions.

Figure D-5. State Transition Diagram, Two-Phase Kill Process

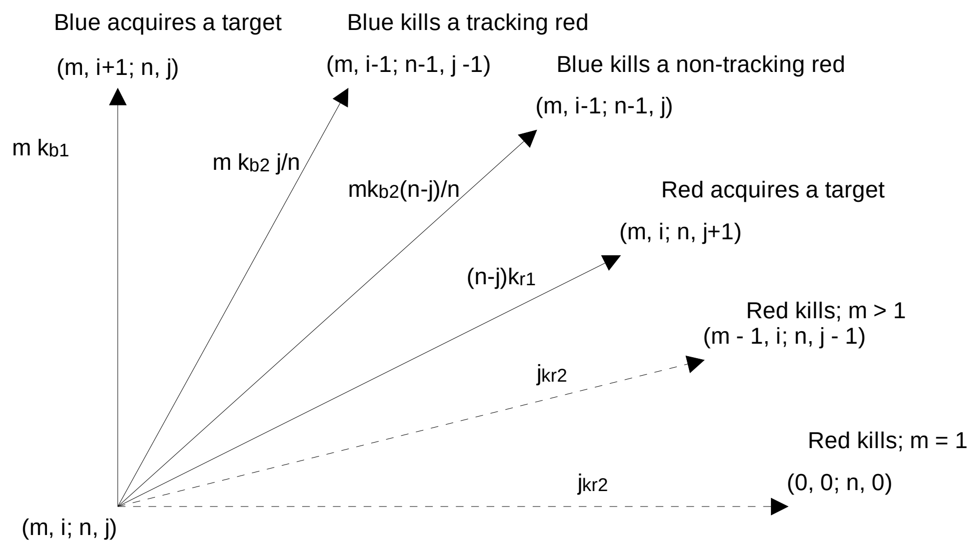


In discussing multiphase kill models and other advanced probabilistic engagement models, we will use dimensioned parameters, because that seems to make the sense of graphs and equations easier to grasp. In Figure D-5, k_{b1} is the parameter of the exponential distribution of the time required for Blue aircraft to lock on to an opponent, and k_{b2} is the parameter of the exponential distribution of the time required for a Blue aircraft to make a kill after lock-on. Parameters k_{a1} and k_{a2} are the homologues of k_{b1} and k_{b2}, respectively, for the Red aircraft. In computing loss distributions, however, we non-dimensionalize the evolution equations with k_{1r}, so that only the three ratios k_{b1}/k_{r1}, k_{b2}/k_{r1}, and k_{r2}/k_{r1} are used, and we consider only long-time limits. Also in Figure D-5, no aircraft can track while scanning, so the rates at which new locks are made is $(m - i)k_{b1}$ and $(n - j)k_{r1}$.

The two-phase kill process allows much more detailed modeling of air-to-air engagements than does the single-phase process. Effects of track-while-scan, of multiple-target tracking, and of sharing targeting information among the elements of a flight can be modeled. Figure D-6 shows how the state transition diagram changes from the one of Figure D-5, when elements of the Blue flights can track

while scanning and share targeting information among flights. The state description for Figure D-6 is (m, i; n, j), where now i is the total number of targets tracked by the Blue flight.

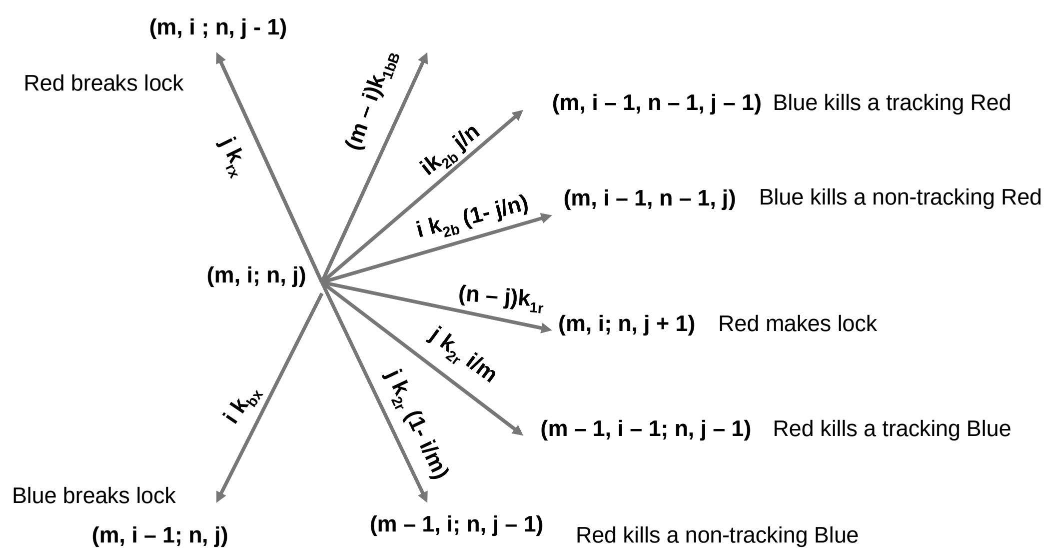
Figure D-6. State Transition Diagram, Track-While-Scan and Information Sharing for Blue Aircraft



m = number of blues, i = number of targets tracked by blue flight;;
n = number of reds,   j = number of tracking reds

The two-phase kill model may be extended to account for a tracked aircraft breaking lock, as shown by the state transition diagram of Figure D-7.

Figure D-7. State Transitions with Lock-Breaking



The evolution equations for the state probabilities are necessarily more complex for two-phase kill processes. For no track-while-scan or sharing of targeting information, they are

$$
\begin{aligned}
\dot{P}_{m,i;n,j} = &-\left[(m-i)kb_1 + ikb_2 + (n-j)kr_1 + jkr_2\right]P_{m,i;n,j} \\
&+[i]^+(m-i+1)kb_1P_{m,i-1;n,j} \\
&+(i+1)kb_2P_{m,i+1,n+1,j}\cdot\frac{n+1-j}{n+1} \\
&+(i+1)kb_2P_{m,i+1,n+1,j+1}\cdot\frac{j+1}{n+1} \\
&+[j]^+(n-j+1)kr_1P_{m,i,n,j-1} \\
&+(j+1)kr_2P_{m+1,i;n,j+1}\cdot\frac{m+1-i}{m+1} \\
&+(j+1)kr_2P_{m+1,i+1,n,j+1}\cdot\frac{i+1}{m+1}
\end{aligned} \qquad . \qquad \text{[Eq. D-2]}
$$

The symbol $[x]^+$ denotes the function of x that equals 1 if $x > 0$, and 0 otherwise.

Though more complex than the differential equations for the evolution of state probabilities with single-phase kill processes, Equation D-2 is still a system of linear ordinary differential equations with constant coefficients. The iterative scheme of Appendix A still allows one to evaluate the long-time limit of absorbing boundary states with a finite-step iterative procedure, and NASA's ASSIST and STEM will treat large-dimensional cases effectively.

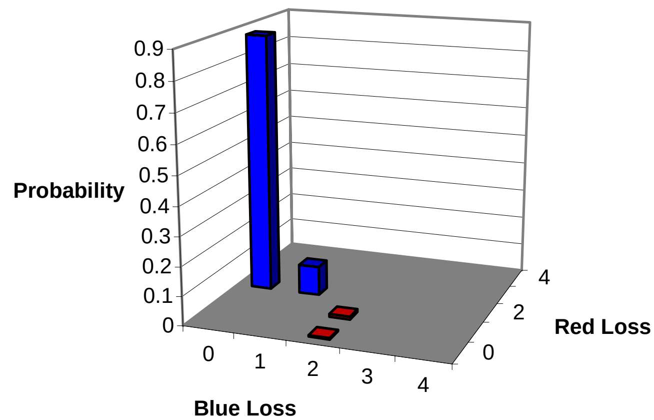
Loss statistics for long-term limits of engagements with the two-phase kill model do not look different from those with the one-phase kill model. They do, however, fit outcomes of certain large-scale simulations better. Table D-1 shows an example in which the three adjustable rate ratios of the two-phase kill model were fitted to give a match of the four outcome probabilities of an EADSIM simulation. The engagement was 4 vs. 4, with both sides breaking at two losses. The fitted rate ratios were $k_{b1}/k_{r1} = 4.94$, $k_{b2}/k_{r1} = 4.90$, and $k_{r2}/k_{r1} = 1.00$.

Table D-1. Loss Probabilities from EADSIM and from Two-Phase Kill Model

Model	(0, 2)	(1, 2)	(2, 1)	(2, 0)
EADSIM	0.886	0.10	0.0084	0.0047
Two-Phase Kill	0.885	0.10	0.01	0.005

Figure D-8 displays the outcome probabilities.

Figure D-8. Loss Probabilities from Two-Phase Kill Model



We were encouraged to find that not only did the outcome probabilities of the two-phase kill model match all four of the EADSIM simulations within about 16 percent, but the time evolution of the absorbing boundary states also agreed reasonably well with those of EADSIM. Figures D-9 and D-10 show this agreement.

Figure D-9. Time Evolution of Probability of Loss State (0,2)

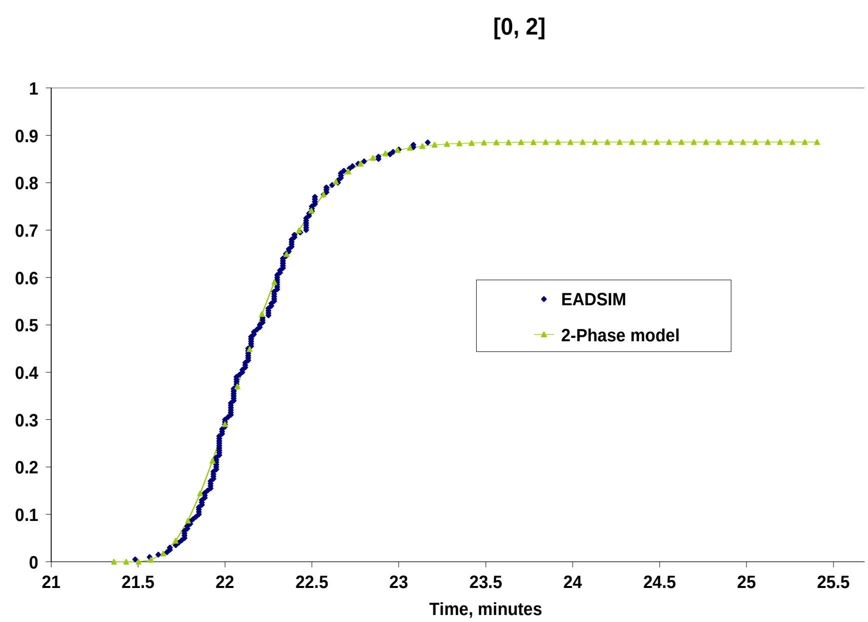
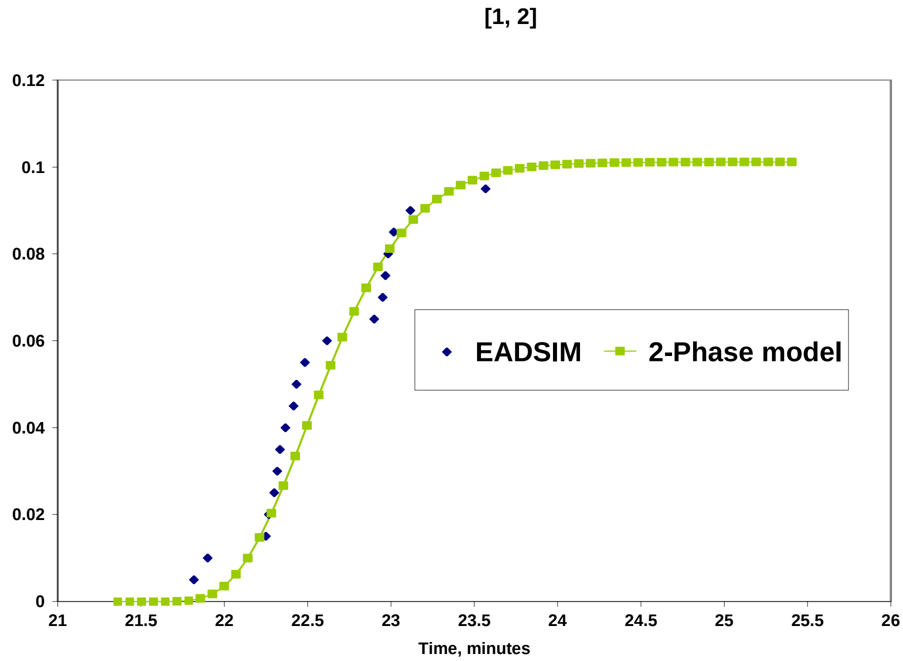


Figure D-10. Time Evolution of Probability of Loss State (1,2)



Since the initial time and time scale are arbitrary in EADSIM, we adjusted those two parameters to give the result in Figure D-10, but made no further adjustments.

MISSILE-TRACKING MODELS

The probabilistic models that we have considered so far implicitly assume that the combatants never run out of missiles. Missile loads may limit aircraft effectiveness, of course, and we have made probabilistic models that account for this. In this section, we describe a probabilistic engagement model in which each aircraft is loaded with 6 missiles.

An engagement state description for this case is (m_0, m_1, …, m_6; n_0, n_1, …, n_6), where m_i denotes the number of Blue aircraft with i missiles, and n_j gives the number of Red aircraft with j missiles.

Table D-2 gives the end states and transition rates for this model, when the starting state is (m_0, m_1, …, m_6; n_0, n_1, …, n_6). In the table, n_{tot} is the total number of Red aircraft, given by $\sum_{j=0}^{6} n_j$, and m_{tot}, computed analogously, is the total number of Blue aircraft. The quantity $u_b = (1 - p_{kb})/p_{kb}$, and $u_r = (1 - p_{kr})/p_{kr}$.

Table D-2. End States and Transition Rates, Missile-Tracking Model

Event	End state	Transition rate
Blue m_i kills a Red n_j, $1 \le i \le 6$, $0 \le j \le 6$	$(m_0, \ldots, m_{i-1} + 1, m_i - 1, \ldots m_6; n_0, \ldots, n_j - 1, \ldots, n_6)$	$m_i\, k_b\, n_j/n_{tot}$
Blue m_i misses, $1 \le i \le 6$	$(m_0, \ldots, m_{i-1} + 1, m_i - 1, \ldots; n_0, n_1, \ldots, n_6)$	$m_i\, k_b\, u_b$
Red n_j kills a Blue m_i, $1 \le j \le 6$, $0 \le i \le 6$	$(m_0, \ldots, m_i - 1, \ldots, m_6; n_0, \ldots, n_{j-1} + 1, n_j - 1, \ldots, n_6)$	$n_j\, k_r\, m_i/m_{tot}$
Red n_j misses, $1 \le j \le 6$	$(m_0, \ldots, m_6; n_0, \ldots, n_{j-1} + 1, n_j - 1, \ldots, n_6)$	$n_j\, k_r\, u_r$

The evolution equations for the state probabilities of the missile-tracking model are again a system of linear ordinary differential equations with constant coefficients, and again the iterative scheme of Appendix A may be used to compute end-state probabilities with finite-step iterations. As Table D-2 suggests, however, the number of equations is large; tens of thousands of transient states would be tracked in a complete model of a 4 vs. 4 engagement in which each aircraft carried 6 missiles, and aircraft left the engagement only by being destroyed. We usually treat missile tracking models with NASA's ASSIST and STEM. Appendix B describes this method.

Loss probability distributions from missile-tracking models have the same general appearance as those from other probabilistic models.

Radar Duel Models

Engagements may not begin with all combatants mutually visible by radar. Rather, they may start with each side, acting on guidance from their own battle management, looking for opponents. The side with the more effective radar, and the smaller radar cross section, may detect opponents before opponents detect them. In this case, they may launch missiles and, seeing these missiles' effects, decide whether or not to continue the engagement. We call this kind of engagement a "radar duel."

In our models of radar duel engagements, the engagement takes place in two phases. In the first phase, the combatants are given guidance from their side's C^4ISR and battle management that is equivalent to an interval of azimuth $\Delta\Theta$ and an interval $\Phi\Delta$, in which to look for their opponents. This phase ends when one side first detects its opponents.

If the side that first detects its opponents has time to fire one or more rounds of missiles and observe their effects before being itself detected, that side may decide either to continue the engagement or to break it off. The second phase ends when the action after the first detection is completed.

Ratios of Detection Ranges

The statistics of detection by pulse-Doppler radars are complex.[1,2] Nevertheless, fairly straightforward considerations, which we summarize here, can relate ratios of detection ranges to properties of radar systems and to battle management.

The signal energy E_s deposited in the receiver of a scanning radar from a target of radar cross section σ at range R in a given azimuth-elevation box is

$$E_s = \frac{\overline{P}G}{4\pi R^2}\frac{\sigma}{4\pi R^2}\frac{A}{L}t_{box}. \qquad \text{[Eq. D-3]}$$

where $\overline{P}$ is the radar's average power, G is the gain of its transmitting antenna, A is the electrical aperture of its receiving antenna, and the factor L accounts for losses. The time t_{box} is the equivalent time that the radar's main lobe dwells in the given azimuth-elevation box.

The time t_{box} is equal to the total time t_{scan} required to cover the search region established by the guidance parameters ΔΘ and ΔΦ, divided by the number of the radar's azimuth-elevation boxes required to cover the search region:

$$t_{box} = t_{scan} / n_{box}. \qquad \text{[Eq. D-4]}$$

The parameter n_{box} will be given roughly by the product of the number of azimuthal beam-widths in the azimuth guidance interval ΔΘ with the number of elevation beam-widths in the elevation guidance interval ΔΦ.[3,4] Thus

$$n_{box} = \frac{\Delta\Theta}{\alpha}\frac{\Delta\Phi}{\beta} \qquad \text{[Eq. D-5]}$$

where α is the radar's azimuthal beam-width and β is its elevation beam-width.[5]

The time t_{scan} is bounded above by the need to complete a scan before the range has changed by a significant fraction of the size of a range box in the pulse-Doppler radar system.

---

[1] Many standard references treat this subject. See, for example, G.W. Stimson, *Introduction to Airborne Radar* (El Segundo, CA: Hughes Aircraft Company, 1983) and M.I. Skolnik, *Introduction to Radar Systems*, Third Edition (New York: McGraw-Hill, 2001).

[2] P. Swerling, "Probability of Detection for Fluctuating Targets," *IRE Transactions on Information Theory*, Volume 6, April 1960, pp. 269-308.

[3] This assumes that battle management's guidance is good enough to position the target in a region that the radar can cover, without turning the aircraft. If that is not so, then detection times increase greatly.

[4] Actual radars may provide fixed choices of angular box widths. For example, the AN/APG-66 provides scans of ±10°, ±20°, or ±30°, with 1, 2, or 4 elevation bars.

[5] See Note 3.

Thus the signal energy available in the receiver is

$$E_s = \frac{\overline{P}G\sigma A}{(4\pi R^2)^2 L} t_{scan} \frac{\alpha}{\Delta\Theta} \frac{\beta}{\Delta\Phi}. \qquad \text{[Eq. D-6]}$$

Noise energy from various sources competes with signal energy. It is customary to characterize all the sources of noise with a single noise temperature T, so that the noise power continuously present in the system is kTB, where B is the system bandwidth in Hz and k is the Boltzmann constant, 1.38×10^{-23} Joule/^oK. The bandwidth B needed for pulse-Doppler processing is close to the inverse of the time on target, $1/t_{ot.}$[6] Thus the noise energy deposited over time t_{ot}, which competes with the signal energy E_s, is kT, and the signal-to-noise ratio SNR is

$$SNR = \frac{\overline{P}G\sigma A}{(4\pi R^2)^2 kT\, L} t_{scan} \frac{\alpha}{\Delta\Theta} \frac{\beta}{\Delta\Phi}. \qquad \text{[Eq. D-7]}$$

Actual detection is probabilistic, and involves consideration of real and false targets. A simple model, which nevertheless captures several features of interest for the radar duel, is to assume that detection is certain when SNR exceeds a critical value SNR*. With this assumption, detection is certain when range R satisfies

$$R \le \frac{1}{2\sqrt{\pi}} \left[\frac{\overline{P}G\sigma A}{SNR^* \, kT\, L} \frac{\alpha}{\Delta\Theta} \frac{\beta}{\Delta\Phi} t_{scan} \right]^{\frac{1}{4}}. \qquad \text{[Eq. D-8]}$$

It follows that the ratio of detection ranges for two combatants a and b is given by

$$\frac{R_a}{R_b} = \left[\frac{\overline{P}_a}{\overline{P}_b} \frac{SNR_b^*}{SNR_a^*} \frac{G_a}{G_b} \frac{T_b}{T_a} \frac{\sigma_b}{\sigma_a} \frac{A_a}{A_b} \frac{L_b}{L_a} \frac{\alpha_a}{\alpha_b} \frac{\beta_a}{\beta_b} \frac{\Delta\Theta_b}{\Delta\Theta_a} \frac{\Delta\Phi_b}{\Delta\Phi_a} \frac{t_{scan\,a}}{t_{scan\,b}} \right]^{\frac{1}{4}}. \qquad \text{[Eq. D-9]}$$

While the detection range ratio of Equation D-9 was derived fairly roughly, it does, we believe, give useful quantitative means of determining which side is likely to win the first phase of a radar duel, from specific information about the two sides' radars and C^4ISR/battle management.

As an example, suppose that the power-aperture product $\overline{P}A$ for the Blue side is 10 times that of the Red side, and that the ratio of the Blues' radar cross section to that of the Red's is 0.01, while all other radar and battle management parameters are the same. Then the ratio of Blue's detection range to Red's, according to Equation D-9 , is 5.6 to 1. Suppose that Blue's detection range is 56 nm. Then Red's detection range is approximately 9 nm.

---

[6] M.I. Skolnik, *Introduction to Radar Systems*, Third Edition (New York: McGraw-Hill, 2001).

Let the sides each have 4 aircraft and be closing at 600 kt. Suppose that, on detecting their adversaries at 56 nm, the Blues develop a firing solution and launch 4 missiles, requiring 30 seconds to do so. When Blue launches, the flights will be 51 nm apart.

Let Blue's missiles fly at Mach 2, 1,200 kt at the altitude flown. If the Reds are moving toward the missiles at 300 kt, the closing rate of the missiles and the Red flight will be 1,500 kt. The missiles will reach their targets in a little over 2 minutes, when the opponents are 31 nm apart, still well outside Red's detection range. If surviving Reds continue toward the Blues, closing at 600 kt, the Blues have 2 minutes to decide whether or not to engage them.

Let us continue this example to determine the distribution of losses in the engagement. Suppose that the 4 missiles that Blue launched at the end of the radar duel phase have a single-shot kill probability of 0.3. Then Red's losses from this phase have a binomial distribution, with outcomes shown in Table D-3.

Table D-3. Red Loss Distribution from Radar Duel Phase

Red aircraft killed	Probability
4	0.0081
3	0.0756
2	0.2646
1	0.4116
0	0.2401

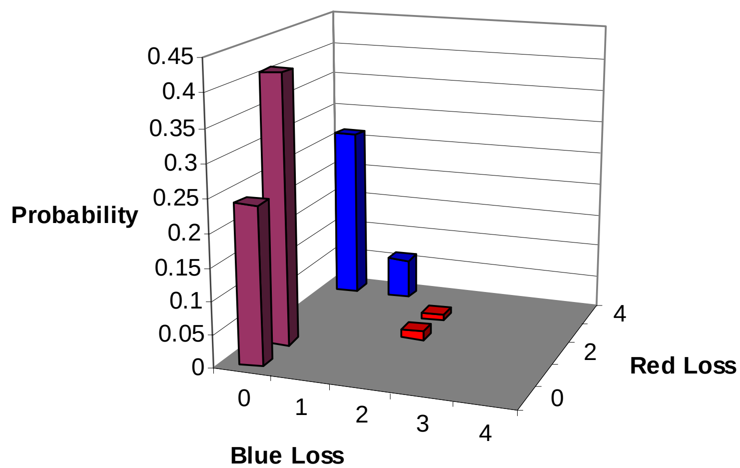
Further suppose that the kill-rate ratio k_b/k_r is 2, and that the Blues' rules of engagement are to continue to a close engagement only if the probability of their sustaining no losses while defeating all their opponents is greater than 70 percent, and that they will break off the near-range engagement on sustaining 2 losses. The M vs. N probabilistic engagement model described above gives the probabilities for zero Blue losses shown in Table D-4. In view of those probabilities, the Blues will continue the engagement only if the Reds lose two or more aircraft in the radar duel phase.

Table D-4. Probabilities for Zero Blue Losses

Engagement	P (no Blue losses)
4 vs. 4	0.345
4 vs. 3	0.517
4 vs. 2	0.711
4 vs. 1	0.889

Modeling those close engagements that occur with the M vs. N probabilistic engagement model, and considering losses from both the radar duel and close-engagement phases, we find the loss distribution of Figure D-11 for this radar duel engagement. In the figure, we show Blue win states with blue bars and Red win states with Red bars. For this example, the outcome (0, 0) is possible. We show the bar for the "tie" outcome in purple. We also show the bar for the outcome (0, 1) in purple, since there seems to be no obvious winner in that case, either.

Figure D-11. Loss Distribution for Radar Duel Example

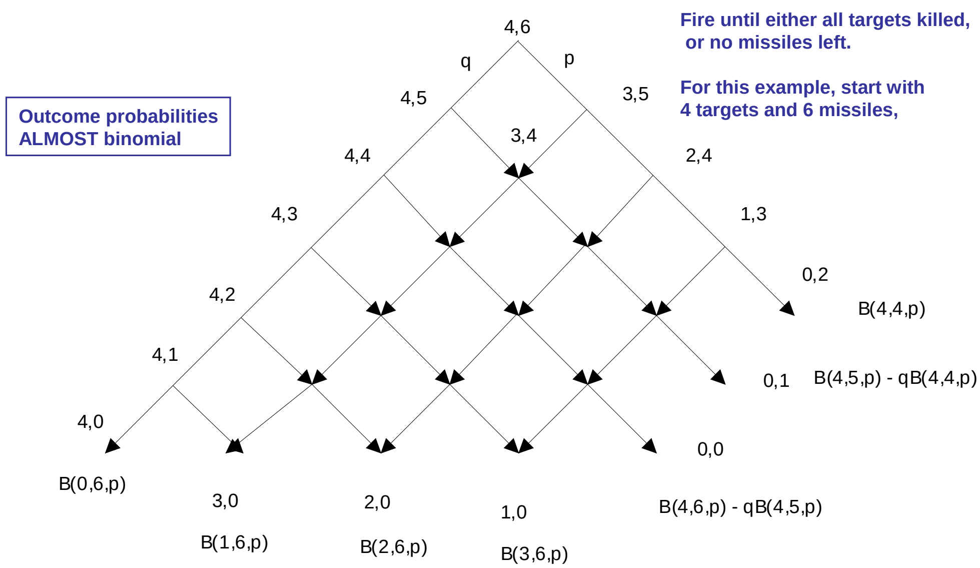


Model for Low-Observable Aircraft

If the probability that Red radars can observe a Blue aircraft is negligibly small, it is reasonable to model that Blue type as invulnerable: it is never destroyed by enemy action in combat. Invulnerable is not the same as invincible, however, because the Blue aircraft still carry only a finite number of missiles, with single-shot kill probabilities less than one.

Engagements with such aircraft against vulnerable foes may be diagrammed and their outcome probabilities computed straightforwardly. Figure D-12 shows an example when invulnerable aircraft with a total of 6 missiles confront 4 vulnerable Blue aircraft (it does not matter how the missiles are distributed among the Blue aircraft). The engagement state is the pair (i, j) where i is the number of enemies and j the number of missiles. The parameter p is the single-shot kill probability of the Blues' missiles, and q is $1 - p$.

Figure D-12. Diagram of Engagement with Invulnerable Blues



In general, if the starting state of such an engagement is (e, m), and e < m, the outcome probabilities are

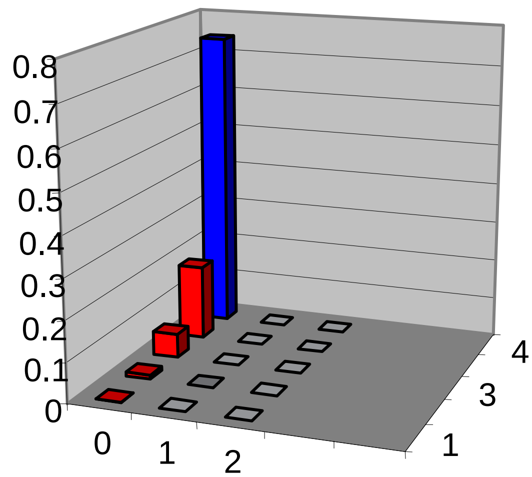
$$
\begin{aligned}
&P(0, m-e-j) = B(e, e+j, p) - qB(e, e+j-1, p), 1 \le j < m-e \\
&P(0, m-e) = B(e, m, p) \\
&P(j, 0) = B(e-j, m, p)
\end{aligned}
\quad . \qquad \text{[Eq. D-10]}
$$

If $m \le e$, all m missiles will of course be fired, and the outcome probabilities are

$$P(e-j, 0) = B(j, m, p), \; 0 \le j \le m . \qquad \text{[Eq. D-11]}$$

Figure D-13 shows a loss distribution for an engagement with invulnerable Blue aircraft. The bars corresponding to Red win states are shown in red. Although no Blues have been lost, they still did not eliminate their opponents before running out of missiles.

Figure D-13. Loss Distribution, Invulnerable Blues



APPLICATIONS TO ANALYSIS

The models considered in this appendix offer means of analyzing the impacts of different features of combat aircraft and missiles in a consistent framework. As an example of this, we give here a consistent treatment of the effects of radars, combat systems, and missile single shot kill probabilities.

We consider a two-phase kill model, in which time-to-lock, a measure of radar effectiveness, is exponentially distributed with parameter k_1, and time-to-fire after lock, a measure of combat system effectiveness, is exponentially distributed with parameter k_2. We denote the missiles' single-shot kill probability by p.

We assume that the firing discipline is lock-shoot-break lock-lock-shoot…, so that the probability q of missing directly influences the time to make a kill. This is different from the effect of p in a single-phase kill process. There, p influences only the rate of firing required to give a specified kill rate.

The mean time to make lock is $1/k_1$, the mean time to fire after making lock is $1/k_2$, and so the mean time to fire is $1/k_1 + 1/k_2$. The probability of making a kill with one shot is p, the probability of making a kill with 2 shots is qp, where $q = 1 - p$, and the probability of requiring n shots to make a kill is $q^{n-1}p$. Accordingly, $\overline{T}$, the expected time to make a kill, is

$$\overline{T} = \sum_{j=1}^{\infty} q^{j-1} p\, j \left[\frac{1}{k_1} + \frac{1}{k_2}\right]. \qquad \text{[Eq. D-12]}$$

Since

$$\sum_{j=1}^{\infty} jq^{j-1} = \frac{1}{(1-q)^2} = \frac{1}{p^2}, \qquad \text{[Eq. D-13]}$$

we have

$$\overline{T} = \frac{1}{p}\left[\frac{1}{k_1} + \frac{1}{k_2}\right] = \frac{1}{p}\left[\overline{T}_L + \overline{T}_F\right], \qquad \text{[Eq. D-14]}$$

where $\overline{T}_L$ is the mean time to make lock, and $\overline{T}_F$ is the mean time to fire after making lock.

Thus all combinations of p, $\overline{T}_L$, and $\overline{T}_F$, for which the quantity ($\overline{T}_L + \overline{T}_F$)/p has a given value, give the same mean time to kill, and Equation D-14 provides a quantitative indication of the way in which radar system effectiveness, combat system effectiveness, and missile effectiveness can be traded.

A COMMENT ON DISTRIBUTIONS

We have used exponential distributions in most of our models. It is certainly appropriate to note that these distributions assign positive probabilities to arbitrarily long times, and that this is not physically realistic because engagements do not last more than a few minutes. Nevertheless, we believe that the exponential distributions are useful, because their logical basis—that the probability of an event occurring in a sufficiently small interval of time is proportional to the size of the interval, and independent of the starting time of the interval, and the probability of simultaneous events is negligibly small—reflects events in the important phases of air-to-air engagements. We would reject conclusions that involved unrealistically long times in any significant way.

The situation is analogous to statistical mechanics. There, too, distributions with unbounded support are used, where extreme values of the support are unrealistic. This leads to such paradoxes as positive, but very small, probabilities that all the molecules in a room would find themselves in one half of the chamber, so that people in the other half would gasp for air. Applied scientists dismiss these predictions, just as we would dismiss predictions involving unrealistically long times.

Over the last 40 years, an extensive literature on stochastic duels has developed, treating engagements (usually with very small numbers of combatants) with arbitrary time-to-kill distributions.[7] If it is essential to use distributions with bounded support, SLAACM can be adapted to use stochastic duel results.

[7] See, for example, A.V. Gafarian and C.J. Ancker, "The Two-On-One Stochastic Duel," *Naval Research Logistics Quarterly*, Vol. 31, 1984, pp. 309–324.

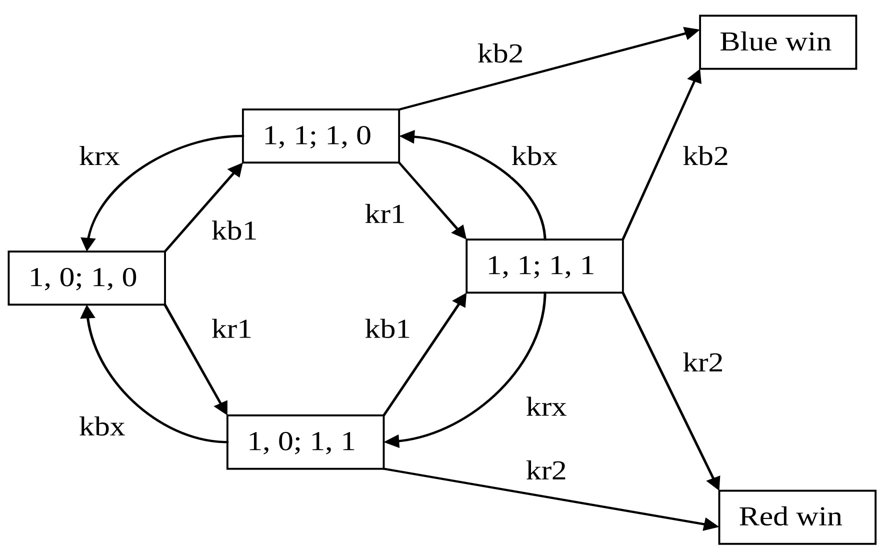
Appendix E
Accounting for Electronic Attack

The method of Appendix A gives a readily-implemented way to allow present SLAACM versions to include impacts of “deception” electronic attacks (EAs). A two-phase kill engagement model with a “break-lock” transition gives a natural way to characterize the impacts of such attacks. Figure E-1 diagrams a 1 vs. 1 engagement, with two-phase kill and break-away.[1] To make a kill, each combatant must make a radar lock on his opponent; having done so, he must fire a (successful) missile. A combatant who perceives that his opponent's radar has locked-on may use deception electronic attack to break the lock.

The system state is (m, i; n, j), where m is the number of Blue aircraft, i the number of Blues having lock; n is the number of Red aircraft, and j is the number of Red aircraft having lock. Labels on the arrows of Figure E-1 show the rates (not probabilities) for each transition.

The parameter kb1 is the reciprocal of the mean time for Blue to make lock, and kb2 is the reciprocal of the mean time for Blue to launch a successful missile, having made lock. Parameter kbx is the reciprocal of the mean time for a Blue aircraft to break a Red lock. The parameters kr1, kr2, and krx have the homologous meanings for Red.

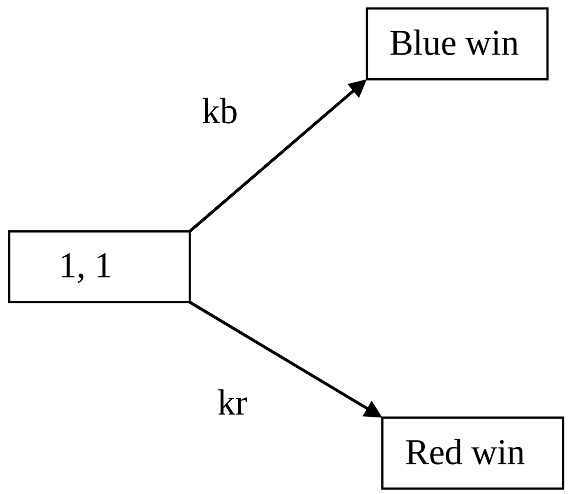
Figure E-1. Diagram of Two-Phase Kill Engagement With Break-Lock



[1] The discussion of the 1 vs. 1 engagement with breakaway includes material from Appendix A, given here for the sake of completeness and ease of reading.

For comparison, Figure E-2 shows a diagram of a 1 Vs 1 engagement with a single-phase kill mechanism. There, the state is simply (number of Blue a/c, number of Red a/c).

Figure E-2. Diagram Of Single-Phase Kill Engagement



Obviously from Figure E-2, for the single-phase kill engagement the probability of a Blue win is equal to kb/(kb + kr), and the probability of a Red win is kr/(kb + kr). The ratio of the probability of a Blue win to the probability of a Red win is, in this case, equal to the kill-rate ratio kb/kr.

This fact suggests inferring an equivalent kill-rate ratio from more complicated 1 vs. 1 engagements, as the ratio of the probability of a Blue win to the probability of a Red win. We now develop equations to do that, for the two-phase kill engagement with break-lock, diagrammed in Figure E-1.

By referring to Figure E-1, one may write down the evolution equations (forward Chapman-Kolmorogov equations) for the engagement. Let $P_{1010}(t)$ denote the probability that the system is in state (1, 0; 1, 0) at time t, and let P_{1110}, P_{1011}, and P_{1111} be the probabilities that the system is in the state indicated by the subscripts. Then, representing the probability of a Blue win at time t by $P_{BW}(t)$ and the probability of a Red win at time t by $P_{RW}(t)$, the evolution equations for the 1 Vs 1 engagement with break-lock are

$$
\begin{aligned}
\dot{P}_{1010} &= -(kb1 + kr1)P_{1010} + krxP_{1110} + kxbP_{1011} \\
\dot{P}_{1110} &= -(kb2 + kr1 + krx)P_{1110} + kb1P_{1010} + kbxP_{1111} \\
\dot{P}_{1011} &= -(kr2 + kb1 + kbx)P_{1011} + kr1P_{1010} + krxP_{1111} \\
\dot{P}_{1111} &= -(kb2 + kr2 + kbx + krx)P_{1111} + kr1P_{1110} + kb1P_{1011} \\
\dot{P}_{BW} &= \quad kb2(P_{1110} + P_{1111}) \\
\dot{P}_{RW} &= \quad kr2(P_{1011} + P_{1111})
\end{aligned}
\quad . \qquad \text{[Eq. E-1]}
$$

In Appendix A we explain how to calculate the long-time limiting values of P_{BW} and P_{RW} by integrating equations F-1 from zero to infinity, using the condition that all the probabilities are zero at t = 0 except P_{1010}, which has the value 1. For completeness, we repeat that development for the case at hand. Defining x_1 as the integral over all positive time of P_{1010}, x_2 as that same integral of P_{1110}, x_3 as that integral of P_{1011}, and x_4 as that integral of P_{1111}, one finds that the x_i are given by the solution of a system of four, linear algebraic equations, the augmented matrix of which is

$$
\begin{pmatrix}
-(kb1+kr1) & krx & kbx & 0 & -1 \\
kb1 & -(kb2+kr1+krx) & 0 & kbx & 0 \\
kr1 & 0 & -(kr2+kb1+kbx) & krx & 0 \\
0 & kr1 & kb1 & -(kb2+kr2+krx+kbx) & 0
\end{pmatrix} .
$$

[Eq. E-2]

One also finds that

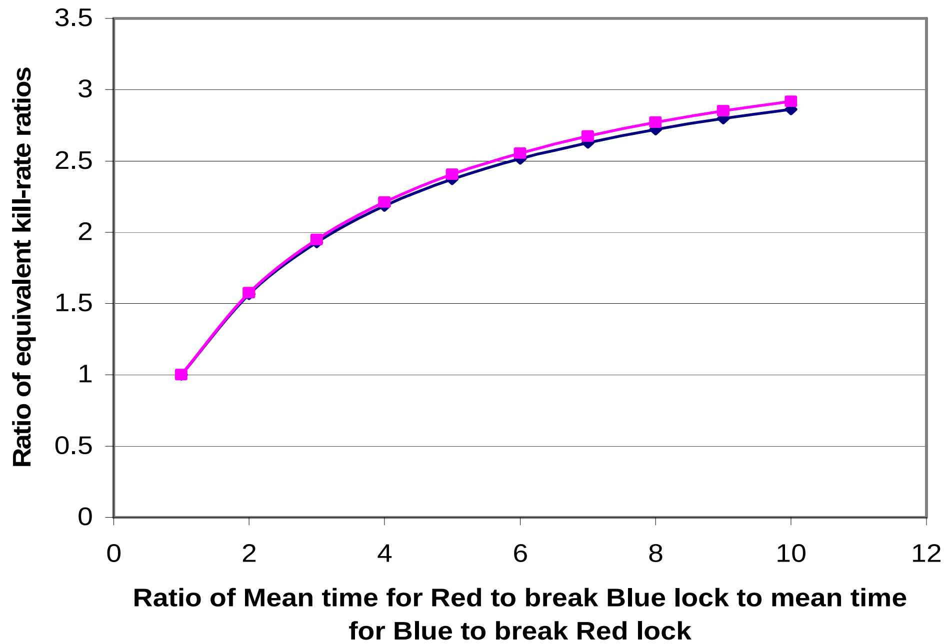
$$
\begin{aligned}
\lim_{t \to \infty} \quad P_{BW} &= kb2(x_2 + x_4) \\
\lim_{t \to \infty} \quad P_{RW} &= kr2(x_3 + x_4)
\end{aligned}
\quad . \qquad \text{[Eq. E-3]}
$$

Given values of kb1, kb2, kbx, kr1, kr2, and krx, the equivalent kill-rate ratio given by P_{BW}/P_{RW} follows directly from Equations E-2 and E-3. Varying the parameters kbx and krx then gives means of inferring the impacts of deception EA on kill-rate ratios.

Numerical experiments suggest that this impact is, in fact (and perhaps fortuitously), fairly straightforward. Figure E-3 shows the variation of equivalent kill-rate ratios with the ratio of kbx to krx, for two cases. In the first, shown with triangle plot points, the mean time for Blue to make lock is 20 seconds, the mean time for Blue to make a kill given lock is three seconds, and the mean time for Blue to break lock is one second; the mean time for Red to make lock is 60 sec-

onds, and the mean time for Red to make a kill given lock is 15 seconds. In the second case, shown with square plot points, the Blue parameters are unchanged, while the mean time for Red to make lock is reduced to 20 seconds, and the mean time for Red to make a kill given lock is three seconds.

Figure E-3. Variation Of Kill-Rate Ratio with Ratio Of Break-Lock Parameter



In both these distinctly different cases, reducing the ratio of the mean time for Red to break lock to the mean time for Blue to break lock from a value of 10 to a value of 1 decreases the equivalent kill-rate ratio by a factor of about 3.

This fact suggests that a crude but useful method of accounting for deception EA by Red aircraft in present SLAACM versions is to allow the user to reduce kill-rate ratios by a factor varying from 1 (no change) to 3 (significant deception EA).

A very much better method would, of course, be to obtain simulation results (TAC BRAWLER or EADSIM, for example) or results from mock combat, with sufficient information to infer effects on kill-rate ratios directly. If, for example, complete outcome distributions for 4 vs. 8 engagements (sample probabilities of i Blue and j Red surviving the engagement) were available, then the five ratios kb1/kr1, kb2/kr1, kbx/kr1, kr2/kr1, and krx/kr1 that characterize two-phase kill engagements with break-lock could be inferred directly for cases with Red EA and without it. These data are not available, however, and until they are, the method proposed here may give useful indications.

Appendix F
Mathematical Analysis: Phased Attack and Defense

In air defense, it is sometimes necessary to provide continuous combat air patrol coverage of the battle space. The current SLAACM includes the option for Blue defense aircraft to be time phased over the battle space in order to provide continuous defense CAP. The inclusion of a phased Blue defense led to discussions with TACAIR about whether Red could phase its attack to take advantage of Blue's phasing. In response to these discussions, we addressed the problem in two substantially different ways, and include both in the sections that follow. Approach 1 demonstrates a method for generating probabilistic results for a well-defined, specific scenario. Note that in this analysis the postulated parameters are selected solely to demonstrate the approach. While these parameters are generally reasonable, they do not represent any known operational scenario. Approach 2 is a generalized combinatorial analysis that explores the probability space for random selections of phasing strategies for both Blue and Red.

APPROACH 1

We are interested in the way CAP attrition is treated in SLAACM. Presently, SLAACM assumes that all the packages Red sends in a "day" are simultaneously observable by Blue's battle management, which may include target identification onboard certain aircraft. This allows certain Blue aircraft to selectively engage high-threat packages. This assumption of high-threat Red attacks seems plausible when saturating Blue's defenses is an attractive option for Red.

But for a scenario in which Blue's target identification/battle management, and the two sides' available forces, would allow Red to saturate Blue's defense with relatively low-value attack packages while Blue remained unaware of upcoming attacks by high-value packages, the SLAACM assumptions are optimistic for Blue. This is particularly true when target identification is onboard aircraft that fly combat air patrols from bases far from the theater.

Full exploration of the options available to each side by time phasing of attacks and defenses will involve considerations of target and basing geographies and may well require case-by-case analysis. Here we consider some specific, simple examples to illustrate the potential significance of these effects.

Suppose that Blue has a fleet of 192 aircraft, which must accomplish a 2-hour flight to the theater. Suppose also that their aircraft can remain on station for 1 hour, that they carry six missiles, that they will keep one missile for self-

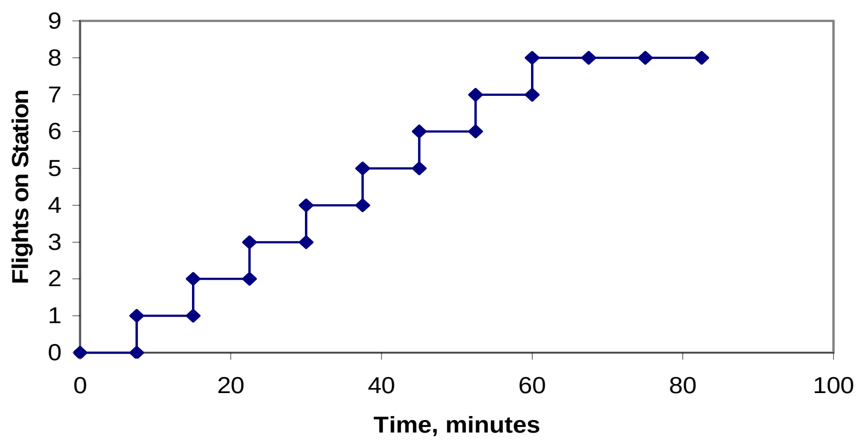
protection while returning to base, and that their turnaround time is 1 hour. This gives the Blue aircraft 1 hour out of six on station, so that six flights must be available to keep one flight continuously on station. Suppose that Blue aircraft always fight in four-ship packages.

Suppose that Red attacks always come in 12-ship packages, made up of advanced escorts, close escorts, and bombers, and that the Blue aircraft must defeat the escorts in order to engage the bombers. Then each 4-ship flight, having 20 usable missiles, can defeat only one attacking package before rearming.

Now, Blue's 192 aircraft provide eight, six-flight groups, so that Blue can keep eight flights on station continuously. In this simple example, Blue controls only the time phasing of the eight groups' CAPs.

Figure F-1 shows an example of the build-up of Blue forces when the CAPs relief times are evenly spaced.

Figure F-1. Arrival Process for Evenly Spaced CAPs



By sending eight attacking packages, Red can at any time "reset" Blue's strength to the start of the buildup, because defeating eight packages exhausts the missiles available to the eight Blue flights on station.[1] In this way the Reds can give themselves intervals of time in which there are fewer than eight Blue flights.

Let us look first at a bad Blue option for the CAPs' relief times. Suppose that, rather than the uniform spacing of Figure F-1, the eight groups' CAPs are exactly in phase. That is, the flights of all eight groups relieve their predecessors at the same time. Then, by sending eight low-value packages, Red can exhaust the missiles available to the Blue force, and there will be no more Blue aircraft for roughly 1 hour. This gives Red considerable scope to dispatch higher-value packages unopposed.

[1] Blue flights might regroup so that the remaining missiles available to two or three flights could be used against another Red package, but this appears to call for complicated reorganization and to give only a second-order effect anyway.

Presumably, Blue will not arrange the groups' CAPs in this way. To consider other Blue options and Red tactics in more detail, we must be more specific about the time required for Red to carry out missions and about the area in which Red packages are vulnerable to Blue's flights.

Let us say, then, that Red's packages are vulnerable for 10 minutes ingress, 5 minutes delivering bombs, and 10 minutes egress. To focus strictly on time phasing, suppose that the Blue defenders are invincible, so that any Blue flight in Red's vulnerable zone, with at least as many Red packages as Blue flights are also in the zone, destroys a Red package.

For further simplification, let us suppose that Red has eight low-value packages and eight high-value packages to dispatch. For specificity, suppose that bombers in the high-value packages carry three times the weight of bombs as those in the low-value packages.

Let us also assume that all eight Blue CAPs are in place, and that the Blue aircraft unfailingly attack high-value packages in preference to low-value packages.

Now, if Red dispatches all 16 packages at once, the eight defending flights will eliminate all eight high-value packages, and Red will deliver eight low-value bomber loads (we'll call this eight "units") of bombs. No aircraft in the low-value packages will be lost.

For completeness, let us treat the above "bad" Blue option in detail. If all eight CAPs were in phase, it seems incredible that Red would not know the times at which the CAPs were relieved. Then Red can send eight low-value packages to arrive just after a relief time. All eight will be destroyed. But the missiles of all eight CAPs will be exhausted too. If the eight high-value packages were undetectable while, say, 20 minutes behind the low-value packages, they could ingress, bomb, and be out of the vulnerable area 15 minutes before the relieving CAPs arrived.

Blue would, however, want to arrange the CAPs' phases to avoid such an outcome. One option would be to space the CAPs' relief points evenly as shown in Figure F-1.

If Red sends the eight low-value packages to arrive just at one CAP's relief time, and delays the eight high-value packages for, say, 20 minutes to make sure that Blue does not know they are coming, the high-value packages will face only two defending flights, with a third defending flight joining after 2.5 minutes. One more Blue flight will arrive just as the five Red packages start bombing; it will destroy a Red package, and four high-value Red packages will drop their bombs.

Two defending flights will arrive before the four packages that survive ingress and bombing leave the vulnerable area (the second arrives just as they leave, but it seems reasonable to give Blue the benefit of a tie), so that the outcome of the attack is six high-value and eight low-value packages destroyed, with four high-value bomber loads of bombs delivered.

Thus, spacing the CAPs' relief times evenly allows Blue to destroy six more high-value packages, and allows four fewer bomber flights to "leak" through Blue's defenses, than would have been the case if the relief times coincided. Red, on the other hand, gets four high-value bomber flights through, delivering 50 percent greater weight of bombs, and loses two fewer high-value packages, than he would have done by sending all the packages simultaneously (as SLAACM now assumes he would do) against the evenly spaced CAPs.

As a final example, we suppose that the eight CAPs' relief times are randomly shifted in time and that Red sends eight low-value packages at a random time. Let us generalize the discussion, assuming that the defending flights' time-on-station is S minutes and that Red follows up eight high-value packages D minutes later. Further suppose that ingress and egress require I and E minutes, respectively, and that bombing takes B minutes.

Then, the number j of Blue defending flights arriving while the high-value packages ingress and drop bombs has the binomial distribution B(j, 8, [D + I + B]/S). The number p of flights that do *not* arrive during that period is, of course, distributed as B(p, 8, [S − D − I − B]/S).

When k defending flights arrive during ingress and bombing, then 8 − k flights arrive in the interval S − I − B, and their arrival times are uniformly distributed over that interval. Thus, the number m of defending flights arriving during egress has the binomial distribution B(m, 8 − k, E/[S − I − B]), for $0 \leq m \leq 8 - k$.

These results give the distribution of the number p of bomber-loads of bombs dropped as B(p, 8, [S − D − I − B]/S), and the distribution P(n) of the total number n of attack packages destroyed as

$$P(n) = \sum_{j=0}^{n} B(n - j, 8, p_1)\, B(j, 8 - n + j, p_2),$$

where $p_1 = (D + I + B)/S$, and $p_2 = E/(S - D - I - B)$. With this information, one can plot the statistics of bomb units dropped and Red packages destroyed (each high-value package delivers three bomb units). Figures F-2 and F-3 show the results for S = 60, D = 20, I = E = 10, and B = 5.

Figure F-2. Probability Distribution of Bomb Units Dropped

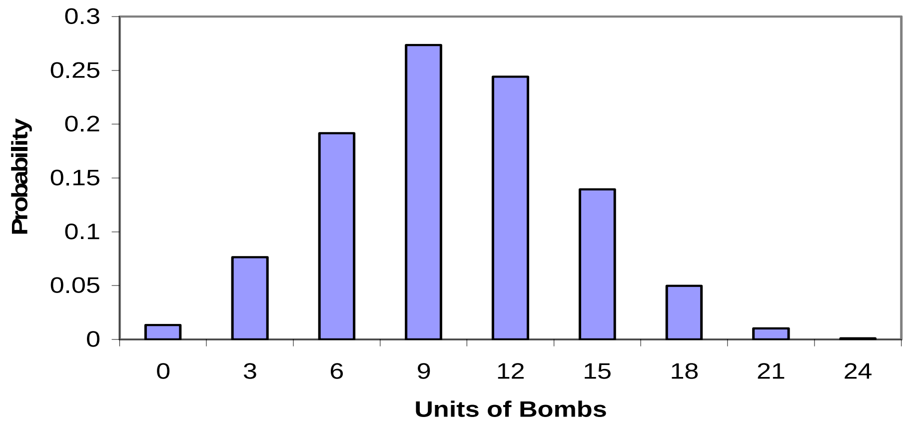
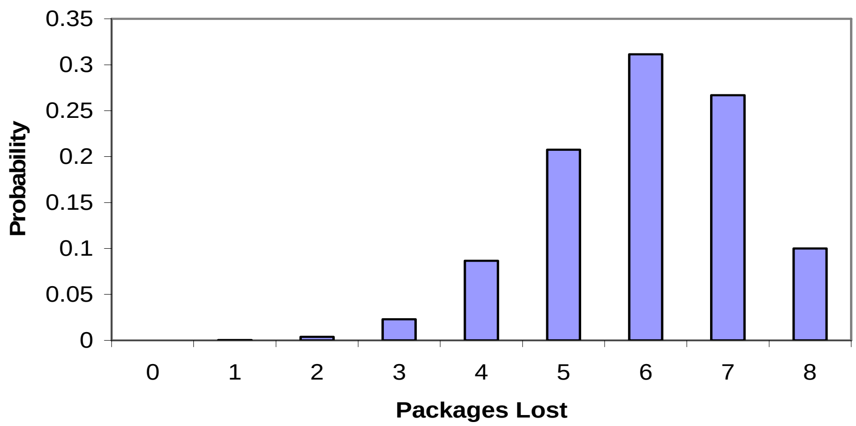


Figure F-3. Probability Distribution of Red Packages Lost



By sacrificing eight low-value packages, Red has 72 percent confidence of delivering more bomb units than with the simultaneous assault now assumed in SLAACM, and a 90 percent confidence of losing fewer high-value packages (these marginal statistics may not, of course, occur together).

APPROACH 2

As discussed above, we are interested in determining if Red can systematically phase his attack to maximize the success of his high-value attack packages. We assume here, as above, that both the Red attack and the Blue defense can be time phased within the engagement period, and each four-ship Blue flight can engage and defeat only one 12-ship Red package. We further assume that the Red attack is numerous, and Red is actually time phasing his high-value packages within a large number of low-value packages, such that Blue defense flights will engage the high-value Red packages available immediately on arrival and will engage the lower value packages if no high-value packages are available. All Blue defense

flights will be absorbed by low-value Red packages whenever high-value packages are not available.

Consider the example in which eight Blue flights can arrive during a single engagement. We assume the engagement is divided into equal periods corresponding to the number of Blue flights, eight in this example. Blue can distribute his flights, and Red can distribute his high-value packages arbitrarily among the eight periods. Because each Blue flight engages only one Red package, Blue can kill (and Red can lose) a maximum of eight high-value Red packages.

We wish to determine if there is a way to phase the Blue arrivals to ensure maximum kills of high-value Red packages. Our assumptions are as follows:

- Blue is invincible and kills Red packages with 100 percent probability.
- Blue flights can preferentially detect and attack high-value Red packages.
- All Blue flights will be committed (or consumed) in each period either by high- or low-value Red packages.
- Red must commit all of his high-value packages during the engagement.

Analysis

By inspection we can deduce that Blue can guarantee at least one high-value package kill by distributing his arrivals equally. Beyond that, the results are not obvious.

To address the Blue and Red options, we need to determine the number of ways Blue and Red can distribute eight flights over eight time slots. This is a multinomial, combination-permutation in which order is important among different quantities, e.g., $(1, 2) \neq (2,1)$, but not important among multiple occurrences of the same quantities, e.g., $(2_1, 2_2) = (2_2, 2_1)$. We first demonstrate a calculation method using four balls in four bins, and later apply the method to eight flights in eight time slots.

Four Balls Distributed Among Four Bins

First, we note that there are 4! ways to distribute four distinguishable items, such as 4 colored balls among 4 bins, 1 to a bin. In our problem, however, the 0, 1, 2, 3, and 4 counts of balls in a bin are distinguishable, but the balls themselves and multiple occurrences of the counts are indistinguishable. The choice of placing 0, 1, 2, 3, or 4 balls in a given bin restricts the options for the remaining bins. For example, if 4 balls are placed in any 1 bin, the rest of the bins must hold 0. (This is same as having 1 red ball and 3 blue balls.) The order of the bin containing the 4 balls is significant, so there are 4 ways to distribute the 4-ball set among the 4 bins. There is only 1! way to arrange the single bin containing 4 indistinguishable balls. If the 0s were distinguishable, there would be 3! ways to distribute them among 3 bins, but we do not distinguish differences in order among the bins containing 0s. When we reduce the

4! maximum options to account for the indistinguishable bins containing the same counts of balls, i.e., the three 0s, we get 4!/(3! * 1!) = 4 unique distributions. Table F-1 confirms this by showing the complete set of 4 unique states.

Table F-1. State Count for Four Balls in a Given Bin

Distribution count	Bins			
	1	2	3	4
1	4	0	0	0
2	0	4	0	0
3	0	0	4	0
4	0	0	0	4

To determine the total number of unique distributions for 4 balls in 4 bins, we need to identify all the positional distributions of unique ball counts. This we do by hand. Table F-2 shows the unique distributions for 4 balls in 4 bins and the corresponding calculations of state counts.

Table F-2. Unique Distributions and State Counts for Four Balls in Four Bins

	Bins				Distribution counts	Distribution counts
	1	2	3	4		
Unique distributions	1	1	1	1	4!/4!	1
	2	1	1	0	4!/(1!*2!*1!)	12
	2	2	0	0	4!/(2!*2!)	6
	3	1	0	0	4!/(1!*1!*2!)	12
	4	0	0	0	4!/(1!*3!)	4
					Total	35

As a check on the method, Table F-3 shows the detailed state enumeration for 4 balls in 4 bins.

Table F-3. State Enumeration for Four Balls in Four Bins

	Bins				Distribution counts
	1	2	3	4	
States	1	1	1	1	1
	2	1	1	0	12
	2	1	0	1	
	2	0	1	1	
	1	2	1	0	
	1	2	0	1	
	0	2	1	1	
	1	1	2	0	
	1	0	2	1	
	0	1	2	1	
	1	1	0	2	
	1	0	1	2	
	0	1	1	2	
	2	2	0	0	6
	2	0	2	0	
	2	0	0	2	
	0	2	2	0	
	0	2	0	2	
	0	0	2	2	
	3	1	0	0	12
	3	0	1	0	
	3	0	0	1	
	1	3	0	0	
	0	3	1	0	
	0	3	0	1	
	1	0	3	0	
	0	1	3	0	
	0	0	3	1	
	1	0	0	3	
	0	1	0	3	
	0	0	1	3	
	4	0	0	0	4
	0	4	0	0	
	0	0	4	0	
	0	0	0	4	
	Total				35

EIGHT FLIGHTS IN EIGHT TIME SLOTS

Now we consider eight flights distributed among eight time slots. The unique distributions and distribution counts are shown in Table F-4.

Table F-4. Unique Distributions and State Counts for Eight Flights in Eight Time Slots

	Time slot								Distribution counts	Distribution counts
	1	2	3	4	5	6	7	8		
Unique distributions	1	1	1	1	1	1	1	1	8!/8!	1
	2	1	1	1	1	1	1	0	8!/(6! 1! 1!)	56
	2	2	1	1	1	1	0	0	8!/(4! 2! 2!)	420
	2	2	2	1	1	0	0	0	8!/(3! 3! 2!)	560
	2	2	2	2	0	0	0	0	8!/(4! 4!)	70
	3	1	1	1	1	1	0	0	8!/(5! 2! 1!)	168
	3	2	1	1	1	0	0	0	8!/(3! 3! 1! 1!)	1,120
	3	2	2	1	0	0	0	0	8!/(4! 2! 1! 1!)	840
	3	3	1	1	0	0	0	0	8!/(4! 2! 2!)	420
	3	3	2	0	0	0	0	0	8!/(5! 2! 1!)	168
	4	1	1	1	1	0	0	0	8!/(4! 3! 1!)	280
	4	2	1	1	0	0	0	0	8!/(4! 2! 1! 1!)	840
	4	2	2	0	0	0	0	0	8!/(5! 2! 1!)	168
	4	3	1	0	0	0	0	0	8!/(5! 1! 1! 1!)	336
	4	4	0	0	0	0	0	0	8!/(6! 2!)	28
	5	1	1	1	0	0	0	0	8!/(4! 3! 1!)	280
	5	2	1	0	0	0	0	0	8!/(5! 1! 1! 1!)	336
	5	3	0	0	0	0	0	0	8!/(6! 1! 1!)	56
	6	1	1	0	0	0	0	0	8!/(5! 2! 1!)	168
	6	2	0	0	0	0	0	0	8!/(6! 1! 1!)	56
	7	1	0	0	0	0	0	0	8!/(6! 1! 1!)	56
	8	0	0	0	0	0	0	0	8!/(7! 1!)	8
									Total	6,435

From Table F-4 we see that there are 6,435 ways to distribute eight Blue flights (and eight Red packages) among eight time slots. Now we want to see the potential impact of this on Blue payoff.

To find the potential payoff for Blue, we want to find the probabilities of Blue (and Red) experiencing 0 through eight flights in a given slot. Table F-5 shows the flight count occurrences for each unique distribution.

Table F-5. Flight Count Occurrences per Unique Distribution

	Unique state distributions, by time slot								Flight count occurrences per unique distribution, by time slot									Distribution counts
	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8	
Unique distributions	1	1	1	1	1	1	1	1		8								1
	2	1	1	1	1	1	1	0	1	6	1							56
	2	2	1	1	1	1	0	0	2	4	2							420
	2	2	2	1	1	0	0	0	3	2	3							560
	2	2	2	2	0	0	0	0	4		4							70
	3	1	1	1	1	1	0	0	2	5		1						168
	3	2	1	1	1	0	0	0	3	3	1	1						1120
	3	2	2	1	0	0	0	0	4	1	2	1						840
	3	3	1	1	0	0	0	0	4	2		2						420
	3	3	2	0	0	0	0	0	5		1	2						168
	4	1	1	1	1	0	0	0	3	4			1					280
	4	2	1	1	0	0	0	0	4	2	1		1					840
	4	2	2	0	0	0	0	0	5		2		1					168
	4	3	1	0	0	0	0	0	5	1		1	1					336
	4	4	0	0	0	0	0	0	6				2					28
	5	1	1	1	0	0	0	0	4	3				1				280
	5	2	1	0	0	0	0	0	5	1	1			1				336
	5	3	0	0	0	0	0	0	6			1		1				56
	6	1	1	0	0	0	0	0	5	2					1			168
	6	2	0	0	0	0	0	0	6		1				1			56
	7	1	0	0	0	0	0	0	6	1						1		56
	8	0	0	0	0	0	0	0	7								1	8
																		6,435

Dividing the flight count occurrences in a given row of Table F-5 by 8 gives the conditional probability of the count occurrences given the distribution corresponding to the row. Dividing the distribution counts by the total count gives the probability of each distribution. Multiplying the distribution probabilities by the conditional count occurrence probabilities generates the probabilities for count occurrences shown in Table F-6.

Table F-6. Flight Occurrences Probabilities for Eight Flights in Eight Time Slots

Distribution probabilities	Flight count probabilities								
	0	1	2	3	4	5	6	7	8
0.00016	0.0000	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.00870	0.0011	0.0065	0.0011	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.06527	0.0163	0.0326	0.0163	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.08702	0.0326	0.0218	0.0326	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.01088	0.0054	0.0000	0.0054	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.02611	0.0065	0.0163	0.0000	0.0033	0.0000	0.0000	0.0000	0.0000	0.0000

Table F-6. Flight Occurrences Probabilities for Eight Flights in Eight Time Slots

Distribution probabilities	Flight count probabilities								
	0	1	2	3	4	5	6	7	8
0.17405	0.0653	0.0653	0.0218	0.0218	0.0000	0.0000	0.0000	0.0000	0.0000
0.13054	0.0653	0.0163	0.0326	0.0163	0.0000	0.0000	0.0000	0.0000	0.0000
0.06527	0.0326	0.0163	0.0000	0.0163	0.0000	0.0000	0.0000	0.0000	0.0000
0.02611	0.0163	0.0000	0.0033	0.0065	0.0000	0.0000	0.0000	0.0000	0.0000
0.04351	0.0163	0.0218	0.0000	0.0000	0.0054	0.0000	0.0000	0.0000	0.0000
0.13054	0.0653	0.0326	0.0163	0.0000	0.0163	0.0000	0.0000	0.0000	0.0000
0.02611	0.0163	0.0000	0.0065	0.0000	0.0033	0.0000	0.0000	0.0000	0.0000
0.05221	0.0326	0.0065	0.0000	0.0065	0.0065	0.0000	0.0000	0.0000	0.0000
0.00435	0.0033	0.0000	0.0000	0.0000	0.0011	0.0000	0.0000	0.0000	0.0000
0.04351	0.0218	0.0163	0.0000	0.0000	0.0000	0.0054	0.0000	0.0000	0.0000
0.05221	0.0326	0.0065	0.0065	0.0000	0.0000	0.0065	0.0000	0.0000	0.0000
0.00870	0.0065	0.0000	0.0000	0.0011	0.0000	0.0011	0.0000	0.0000	0.0000
0.02611	0.0163	0.0065	0.0000	0.0000	0.0000	0.0000	0.0033	0.0000	0.0000
0.00870	0.0065	0.0000	0.0011	0.0000	0.0000	0.0000	0.0011	0.0000	0.0000
0.00870	0.0065	0.0011	0.0000	0.0000	0.0000	0.0000	0.0000	0.0011	0.0000
0.00124	0.0011	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002
Totals	0.4667	0.2667	0.1436	0.0718	0.0326	0.0131	0.0044	0.0011	0.0002

Table F-7 repeats the occurrence probabilities from Table F-6 and adds the Blue payoff probabilities, which are calculated by multiplying the probabilities by their corresponding flight counts. The expected payoff shown in Table F-7 is 1 compared to the maximum payoff of 8.

Table F-7. Probabilities and Payoff for Eight Flights in Eight Time Slots

Probability per state									
0	1	2	3	4	5	6	7	8	Total
0.4667	0.2667	0.1436	0.0718	0.0326	0.0131	0.0044	0.0011	0.0002	1.000

Blue payoff/Red loss probability									
0	1	2	3	4	5	6	7	8	Total
0	0.2667	0.2872	0.2154	0.1305	0.0653	0.0261	0.0076	0.0012	1.000

Look-Ahead Target Identification

The case in which Blue can see beyond the current period, and thus avoid wasting flights on low-value packages, can be modeled simply by reducing the number of time slots. Using the same approach followed above, we find that the expected payoff for eight flights in four time slots is 2, and the expected value for eight flights in two time slots is 4. The simplicity of these numbers suggests that there may be a more fundamental way to derive them than we have applied. That said,

the results demonstrate the value of long-range identification to Blue and the value of preventing such identification to Red.

Approach 2 Summary

We note that these results represent the case in which Blue and Red each randomly select one of an exhaustive set of strategies. *It does not represent the case in which Blues and Reds arrive randomly*. The case of random phasing of Blue arrivals, and random arrival of the Red attack, is considered in Approach 1.

At this time, we do not know of any gaming strategy that can reliably improve the results for either Blue or Red.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(MM-YYYY)*
05-2007

2. REPORT TYPE
Final

3. DATES COVERED *(From – To)*

4. TITLE AND SUBTITLE
Stochastic Lanchester Air-to-Air Campaign Model

Methods Used to Generate Model Outputs
and a Users Guide: 2007

5a. CONTRACT NUMBER
GS-00F-0026M

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Hemm, Robert V.; Author
Lee, David A.; Author
Eckhause, Jeremy M.; Author
Dukovich, John A.; Author

5d. PROJECT NUMBER

5e. TASK NUMBER
PA603

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

LMI
2000 Corporate Ridge
McLean, VA 22102-7805

8. PERFORMING ORGANIZATION REPORT NUMBER
LMI-PA603T2

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
OSD/Program Analysis and Evaluation - TACAIR
Mr. Frank Lewis
Pentagon, Room 2C281
Washington, DC 20301-1800

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
A Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The report describes latest version of the Stochastic Lanchester Air-to-Air Campaign Model (SLAACM). The report has two major sections and appendices. The first section includes several chapters that describe the mathematics and analysis methods used for engagement modeling, bomb damage calculations, campaign planning, and integer program optimization. The second section is a user's guide for the latest SLAACM release. The appendices describe the unique and efficient engagement algorithm used in SLAACM, supporting Markov analysis tools, mathematics to support additional combat scenarios, and results of supplemental analyses.

15. SUBJECT TERMS
Multiple air combat, campaign models, Lanchester models

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Nancy E. Handy
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	Unclassified Unlimited	58	**19b. TELEPHONE NUMBER** *(include area code)* 703-917-7249

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18